C

AN OPTIMAL PARAMETER DISCRETIZATION STRATEGY FOR MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL

DISSERTATION

Stuart N. Sheldon
Captain, USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

1

AFIT/DS/ENG/89-2

AN OPTIMAL PARAMETER DISCRETIZATION STRATEGY FOR MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL

DISSERTATION

Stuart N. Sheldon
Captain, USAF

AFIT/DS/ENG/89-2

Approved for public release; distribution unlimited

AFIT/DS/ENG/89-2

AN OPTIMAL PARAMETER DISCRETIZATION STRATEGY FOR MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL

DISSERTATION

Presented to the Facult: of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Stuart N. Sheldon, B.S.M.E., M.S.
Captain, USAF

December 1989

Approved for public release; distribution unlimited

AFIT/DS/ENG/89 2

AN OPTIMAL PARAMETER DISCRETIZATION STRATEGY FOR MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL

Stuart N. Sheldon, B.S.M.E., M.S.
Captain, USAF

Approved:

Peter S. Maybeck 22 Nov 89
Peter S. Maybeck, Chairman

Vadim Komkov 22 Nov 89
Vadim Komkov

Michael B. Leahy Jr. 22 NOV 89
Michael B. Leahy

Accepted:

Robert A Calico Jr
Robert A. Calico, Jr.
Interim Dean, School of Engineering

Preface

This dissertation presents the first formal design method for finite bank multiple model adaptive estimators and controllers for systems with continuously variable parameters. Although it seems like such an easy solution, one must remember that before this research was accomplished, there was great confusion as to what parameters to choose for the design.

I would like to thank my research committee; Dr. Peter Maybeck, Capt. Mike Leahy, and Dr. Vadim Komkov, for helping me form this dissertation into a viable document. I would expressly like to thank Dr. Maybeck for his confidence in my work and for his thorough review at each step in my research. This proved to be the motivation for me to complete this dissertation. I would be remiss if I did not thank my office mates; Tim Kearns, Randy Paschall, Paul Bailor, Bruce Conway, Ron Jackson, and Jeff Simmers, for providing pleasant diversion from the day to day workload with endless conversation. I would finally like to thank my wife, Marla, and children, Stuart and Kevin, for helping me to remember what life is all about.

> "Yes, yes, I *know* that, Sidney ... everybody knows *that!* ... But look: Four wrongs squared, minus two wrongs to the fourth power, divided by this formula, *do* make a right." *Gary Larson*

Stuart N. Sheldon

Table of Contents

Page

List of Figures

List of Tables

List of Symbols

Symbol	Page Defined
a_i	6
$\mathbf{a}^*$	2
$\mathbf{a}^k$	2
$\hat{\mathbf{a}}$	4
$\mathbf{A}_p$	35
$\mathcal{A}_i$	6
$\mathbf{B}(t)$	55
$\boldsymbol{B}_d^k(t_i)$	41
$\boldsymbol{B}_d^*(t_i)$	41
$C^k(t_i)$	42
$C^*(t_i)$	42
$\mathrm{E}\{\bullet\}$	9
$\mathbf{F}(t)$	55
$\boldsymbol{G}_o$	25
$\boldsymbol{G}_a$	36
$\boldsymbol{G}_c^k$	43
$\boldsymbol{G}_d^k(t_i)$	22
$\boldsymbol{G}_d^*(t_i)$	22
$\mathcal{G}_0$	45
$\mathcal{G}_a$	52
$\boldsymbol{H}^k(t_i)$	22
$\boldsymbol{H}^*(t_i)$	22
J^2	10
J_a^2	10

Symbol	Page
J_c^2	12
J^∞	9
J_o	59
$\mathcal{J}$	91
K	2
$\mathcal{L}$	26
ℓ^k	29
m	22
$\boldsymbol{M}^k(t_i)$	3
n^k	21
n^*	22
$\boldsymbol{N}^k$	29
p	41
$p^k(t_i)$	2
q	42
$\boldsymbol{Q}_o$	25
$\boldsymbol{Q}_d^k$	22
$\boldsymbol{Q}_d^*$	22
$\mathbf{r}^k(t_i)$	3
$\boldsymbol{R}^k$	22
$\boldsymbol{R}^*$	23
s^k	22
s^*	22
$\mathbf{S}$	91
$\boldsymbol{T}$	25
$\mathcal{T}$	45
$\mathcal{T}_a$	52

Symbol	Page
$\mathbf{u}^*$	5
$\mathbf{u}^k$	5
$\mathbf{U}$	92
$\mathbf{v}^k(t_i)$	22
$\mathbf{v}^*(t_i)$	23
$\mathbf{w}_d^k(t_i)$	22
$\mathbf{w}_d^*(t_i)$	22
$\mathbf{W}$	9
$\mathbf{x}_a(t_i^-)$	35
$\mathbf{x}^k(t_i)$	21
$\mathbf{x}^*(t_i)$	22
$\hat{\mathbf{x}}$	4
$\hat{\mathbf{x}}^k(t_i^-)$	24
$\hat{\mathbf{x}}^k(t_i^+)$	24
$\tilde{\mathbf{x}}^k$	26
$\mathbf{y}(t_i)$	12
$\mathbf{y}^k(t_i)$	42
$\mathbf{y}^*(t_i)$	42
$\mathbf{z}^k$	22
$\mathbf{z}^*$	22
$\mathbf{Z}_i$	2
$\mathbf{Z}(t_i)$	2
$\boldsymbol{\Gamma}_i^k$	26
ζ	56
λ	10
$\boldsymbol{\Xi}_i^k$	25
$\boldsymbol{\Upsilon}$	25

Symbol	Page
$\boldsymbol{\Upsilon}_a$	36
φ	49
$\boldsymbol{\Phi}^k(t_{i+1}, t_i)$	22
$\boldsymbol{\Phi}^*(t_{i+1}, t_i)$	22
$\boldsymbol{\Psi}$	9
$\boldsymbol{\Psi}_a$	35
$\boldsymbol{\Psi}_c$	40
$\boldsymbol{\Psi}_o$	58
$\boldsymbol{\Psi}^{k-}$	28
ω_n	56

AFIT/DS/ENG/89-2

Abstract

A method is proposed for designing multiple model adaptive estimators to provide combined state and parameter estimation in the presence of an uncertain parameter vector. It is assumed that the parameter varies over a continuous region and a finite number of constant-gain filters is available for the estimation. The estimator elemental filters are chosen by minimizing a cost functional representing the average state prediction error autocorrelation, with the average taken as the true parameter ranges over the admissible parameter set. A second-order example is used to illustrate the increase in performance over previously accepted filter selection methods. By minimizing a cost functional representing the average parameter prediction error autocorrelation, a parameter estimator is designed which is different from the state estimator. The parameter estimator found with this method provides lower average mean square parameter estimation error than previously accepted design methods.

An analogous method is proposed for designing multiple model adaptive controllers to provide stabilizing control in the presence of an uncertain parameter vector. A finite number of constant-gain controllers is used to regulate a system with a parameter vector that varies over a continuous region of the parameter space. The controller elemental filters are chosen by minimizing a cost functional representing the average regulation error autocorrelation, with the average taken as the true parameter ranges over the admissible parameter set. A second-order example is used to demonstrate the improvement in performance over previously accepted controller selection methods. Modifications to the estimator and controller optimization algorithms are investigated which provide the additional capabilities of: changing the filter selection mechanization, using different types of cost functionals, and placing lower bounds on the model probabilities to keep the system open for adaptation.

AN OPTIMAL PARAMETER DISCRETIZATION STRATEGY FOR MULTIPLE MODEL ADAPTIVE ESTIMATION AND CONTROL

I. Introduction

1.1 Overview

This thesis presents an area of research in the design of adaptive estimators and controllers. It centers around *multiple model adaptive estimation* (MMAE), using a bank of Kalman filters for estimating the states and parameters of a system. MMAE is used as the basis for *multiple model adaptive control* (MMAC) systems which provide control in the presence of parameter variation. Section 1.2 provides some background on MMAE and MMAC.

The design method proposed herein involves optimally choosing the models upon which to base the elemental filters of MMAE systems, or the elemental controllers of MMAC systems, with parameters that can vary over a continuous range. The problem is choosing a finite subset of representative parameters from an infinite parameter set. Section 1.3 describes the model selection problem encountered when using MMAE that is solved through this research.

Section 1.4 outlines the scope of this research effort through the statement of three hypotheses which are supported in the following chapters. Section 1.5 outlines the approach taken in this research effort and provides some direction for developing similar design methods. Section 1.6 presents the major assumptions which facilitate this method and outlines the approximations made in the theoretical development.

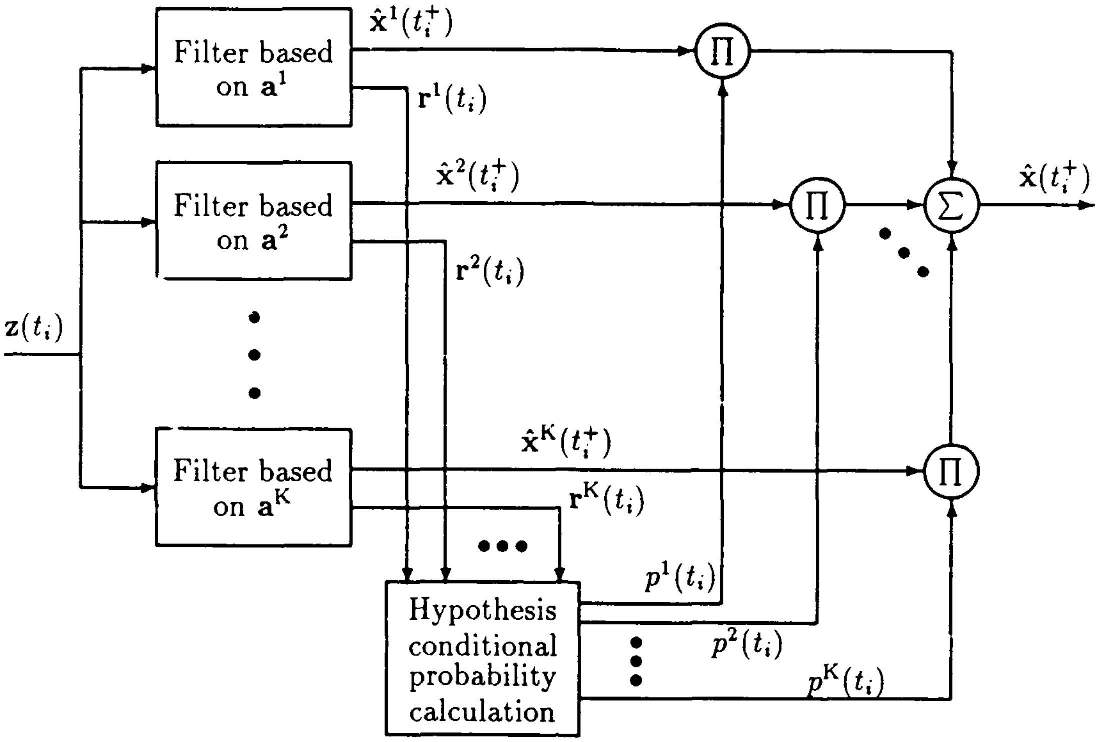
1.2 Background

MMAE is accomplished through the use of a *multiple model filtering algorithm* (MMFA) [1, 6, 33, 36]. The MMFA has been shown to perform well as a filter for tracking airborne targets [7, 12, 31, 32, 38, 41, 46, 49, 55, 59, 60, 61], for tracking submarines [47], for estimating parameters in space structure control systems [10, 23, 24, 29, 30], for estimating parameters in flight control systems [3, 5, 34, 54], for estimating the position of a neutral particle beam [15, 22, 42, 43, 44, 65], for estimation in a terrain assisted navigation system [56], and for estimation in a tank fire control system [64]. The AFIT robotics research group proposes using MMAE to estimate the changing parameters in a robotic manipulator [57, 58]. This research is concerned with choosing the models in an MMFA to minimize the mean squared estimation error.

1.2.1 Multiple Model Adaptive Estimation. The MMFA is an algorithm originally proposed by Magill [33] to provide accurate state estimation under parameter variation. The true parameter vector, $\mathbf{a}^*$, is assumed to be a member of a finite set of possible vectors, $\{\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^K\}$. As shown in Figure 1, a bank of Kalman filters is constructed with each filter based upon the hypothesis that $\mathbf{a}^* = \mathbf{a}^k$, $k = 1, 2, \ldots, K$. The filters run in parallel, and the residuals in each filter are monitored by an executive routine. This routine calculates the hypothesis conditional probability, $p^k(t_i) \triangleq \text{prob}\{\mathbf{a}^* = \mathbf{a}^k \mid \mathbf{Z}(t_i) = \mathbf{Z}_i\}$, for each of the elemental filters, where$\mathbf{Z}_i$ represents a realization of the measurement history,$\mathbf{Z}(t_i)$, which itself is composed of partitions that are the individual measurements at each sample time, $\mathbf{z}(t_1), \mathbf{z}(t_2), \ldots, \mathbf{z}(t_i)$.

The conditional densities are used to update this probability at each sample instant by the calculation [53]:

$$p^k(t_i) = \frac{f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}^k, \mathbf{Z}_{i-1})p^k(t_{i-1})}{\sum_{j=1}^{K} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}^j, \mathbf{Z}_{i-1})p^j(t_{i-1})} \qquad (1)$$



Filter based on $\mathbf{a}^1$
Filter based on $\mathbf{a}^2$
Filter based on $\mathbf{a}^K$
$\mathbf{z}(t_i)$
$\hat{\mathbf{x}}^1(t_i^+)$
$\mathbf{r}^1(t_i)$
$\hat{\mathbf{x}}^2(t_i^+)$
$\mathbf{r}^2(t_i)$
$\hat{\mathbf{x}}^K(t_i^+)$
$\mathbf{r}^K(t_i)$
Π
Σ
$\hat{\mathbf{x}}(t_i^+)$
Hypothesis conditional probability calculation
$p^1(t_i)$
$p^2(t_i)$
$p^K(t_i)$

Figure 1. Multiple Model Filtering Algorithm

where the conditional density for the current measurement, $\mathbf{z}(t_i)$, is given by:

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}^k, \mathbf{Z}_{i-1}) = \frac{1}{(2\pi)^{m/2} \mid \boldsymbol{M}^k(t_i) \mid^{1/2}} \exp\{-\frac{1}{2}\mathbf{r}^{k\mathrm{T}}(t_i)[\boldsymbol{M}^k(t_i)]^{-1}\mathbf{r}^k(t_i)\} \tag{2}$$

where$\mathbf{r}^k(t_i)$ is the residual vector and $\boldsymbol{M}^k(t_i)$ is the anticipated residual covariance generated in the k^{th} Kalman filter as

$$\boldsymbol{M}^k(t_i) = \boldsymbol{H}^k(t_i)\boldsymbol{P}^k(t_i^-)\boldsymbol{H}^{k\mathrm{T}}(t_i) + \boldsymbol{R}^k(t_i) \tag{3}$$

The initial probability distribution $p^k(t_0)$ is chosen based on any prior information available. For example, if each model is equally likely, the a priori distribution could be set to $p^k(t_0) \triangleq 1/\mathrm{K}$, $k = 1, 2, \ldots, \mathrm{K}$.

There are two accepted methods to calculate state and parameter estimates once one has the conditional probability distribution. The Bayesian method produces a state estimate which is the conditional mean of the state vector $\mathbf{x}$. The conditional mean of $\mathbf{x}$ is

$$\hat{\mathbf{x}}(t_i^+) \triangleq \sum_{k=1}^{K} \hat{\mathbf{x}}^k(t_i^+)p^k(t_i) \tag{4}$$

where $\hat{\mathbf{x}}(t_i^+)$ denotes the estimate at time t_i after the measurement update. The term $\hat{\mathbf{x}}(t_i^-)$ is used in the following chapter to denote the estimate at time t_i before the measurement update. The parameter estimate, the conditional mean of $\mathbf{a}$, can be found, if desired, by calculating

$$\hat{\mathbf{a}}(t_i) \triangleq \sum_{k=1}^{K} \mathbf{a}^k p^k(t_i) \tag{5}$$

The *maximum a posteriori* (MAP) method chooses the state and parameter estimates as the values in the model with the highest probability:

$$\hat{\mathbf{x}}(t_i^+) \triangleq \hat{\mathbf{x}}^k(t_i^+) \text{ for } k \text{ such that } p^k(t_i) = \max_j\{p^j(t_i)\} \tag{6}$$

A major assumption underlying this research is that the MMFA converges to the proper model. Baram showed in his dissertation [4] that both the Bayesian and MAP methods converge in steady state to the model closest to the actual parameter value with respect to his information measure. Therefore, this research is applicable to either of these methods of model selection, though in any particular implementation, either method has advantages and disadvantages. The MAP method chooses the model which is most probably the correct filter. However, as the selected filter changes, the state estimate has jump discontinuities. The Bayesian method has much smoother transitions between filters as the parameter changes. However, the state estimate has a contribution from filters with a low probability of being the correct filter.

In this research, the MMFA is restricted to have steady-state constant-gain Kalman filters. That is, each filter will be based on a time invariant model with stationary noises, and the precalculated steady state value of the covariance matrix will be used to calculate the gain, thus ignoring any natural initial transient. This

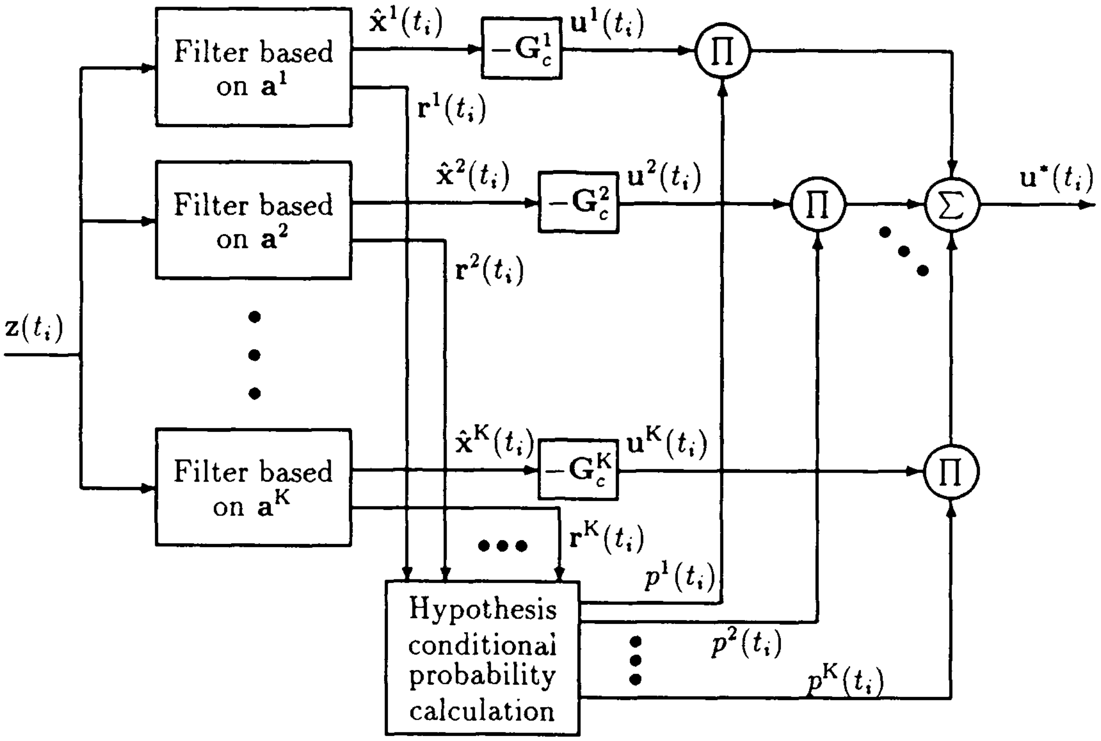


Figure 2. Multiple Model Control Algorithm

also implies that the steady-state value of $\boldsymbol{M}^k$ is used in the conditional density of Equation (2).

1.2.2 Multiple Model Adaptive Control. MMAC is accomplished through the use of a ***multiple model control algorithm*** (MMCA) [3, 13, 14, 27, 37]. The MMCA is an extension of the MMFA in which each elemental filter is appended with a gain to form the elemental controllers as shown in Figure 2. These state feedback control laws are calculated for each model by any suitable method, such as LQG synthesis or eigenstructure assignment. The control to the plant is then calculated as an average of the control signals, weighted by the conditional probability as follows:

$$\mathbf{u}^*\,(t_i) \triangleq \sum_{k=1}^{K} \mathbf{u}^k\,(t_i)p^k(t_i) \tag{7}$$

Aside from the publications deriving and explaining the multiple model formulations [1, 6, 8, 20, 26, 27, 33, 45], research has been done on convergence of the estimator [4, 17, 18], setting bounds on the error of the estimator [17, 18], analysis of the oscillation phenomena in an MMCA [13, 14], estimation of distributed systems [21], decentralized estimation and control [16], moving bank structures [19, 23, 24, 29, 30, 39], and methods for choosing the representative models [24, 28, 30, 34, 39, 52].

1.3 Problem Statement

The theoretical development of the MMFA assumes the true parameter vector

$$\mathbf{a}^* \triangleq \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_r \end{bmatrix}$$

is a member of a finite set of possible vectors, $\{\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^K\}$. In most physical systems, each parameter varies over a continuous bounded range

$$a_i \in \mathcal{A}_i = [a_i^l, a_i^u] \quad \text{for} \quad i = 1, 2, \ldots, r$$

which is called the admissible set of parameter values. This means there is an infinite number of possible values for each parameter. It is impossible to design a bank containing an infinite number of filters, so a finite subset of parameters must be chosen from this set. The models in the MMFA or MMCA are based on this finite subset of parameter values.

To make MMAE and MMAC effective and attractive to the practicing engineer, there needs to be a practical method for choosing the finite subset which ensures that it represents the admissible set of parameter values. A subset of parameters is considered a representative subset if an MMFA or MMCA based on this subset performs adequately when the true parameter vector is anywhere in the admissible set. The question "What is adequate?", leads to the definition of the problem:

Problem: *Previously accepted methods of choosing representative parameters from the admissible set of parameter values, for an MMFA or MMCA, are ad hoc and do not provide a means for measuring the quality of the designer's choice of models.*

1. Sengbush & Lainiotis proposed in [52] two algorithms for binary quantization of the parameter space.
2. Lamb & Westphal used a simplex method of nonlinear programming in [28] to direct the parameter selection.
3. Matthes used performance boundaries from simulations to define the subset covered by each of his models, then chose enough models to cover the entire set of interest in [34].

In applications research at AFIT, several methods of model selection have been implemented. One of the simplest methods is to divide the parameter set linearly so the models are an equal distance apart in the parameter space [19, 23, 24, 39]. Equal spacing on a logarithmic scale has been used for estimating a break frequency parameter [19, 39]. When this method did not provide good results, Lashlee [29, 30] defined a nominal model and chose neighboring models based on the amount of performance degradation experienced when the true parameter value was that of a neighboring model. For tracking problems, models were chosen based on the dynamics expected from the target [41, 55, 60, 61].

These results and the work done by Matthes [34] point toward using the performance of the estimator (and/or controller) as the best choice of model selection criterion. Therefore, this research is aimed toward developing a method, based on estimation (control) performance, for optimizing the model selection for a MMFA (MMCA) to provide the best estimation (control) with the least number of models. The primary goal of this research has been the development of an optimization procedure for choosing the discretization of a parameter space. This entails defining a measure of quality for a particular choice (i.e., a cost functional to minimize) and choosing a method to accomplish the minimization. There are two approaches that would be beneficial to a design engineer:

1. When faced with limitations on the amount of computing power available, it would be desirable to choose the parameter discretization which would provide the lowest estimation or tracking error based on a prespecified number of models.
2. When faced with constraints on performance, it would be desirable to choose the discretization based on the amount of estimation or tracking error that can be tolerated. The idea would be to choose the fewest number of model parameters from the parameter set to achieve at least the desired performance.

The next section sets forth limits on the scope of this research so that the primary goal is achieved through focused effort.

1.4 Scope

The scope of this effort is the development of relationships and methods to support the following hypotheses:

Hypothesis 1a *There exists a representative subset of the admissible parameter set, which forms the basis of an MMFA, of prespecified size, to provide state estimation with minimum estimation error.*

Hypothesis 1b *There is a minimum number of parameters in any representative subset of the admissible parameter set, forming the basis of an MMFA, that provide state estimation with a prespecified bound on the estimation error.*

Hypothesis 2a *There exists a representative subset of the admissible parameter set, which forms the basis of an MMFA, of prespecified size, to provide parameter estimation with minimum estimation error.*

Hypothesis 2b *There is a minimum number of parameters in any representative subset of the admissible parameter set, forming the basis of an MMFA, that provide parameter estimation with a prespecified bound on the estimation error.*

Hypothesis 3a *There exists a representative subset of the admissible parameter set, which forms the basis of an MMCA, of prespecified size, to provide control with minimum tracking error.*

Hypothesis 3b *There is a minimum number of parameters in any representative subset of the admissible parameter set, forming the basis of an MMCA, that provides control with a prespecified bound on the tracking error.*

These hypotheses all deal with finding a system which provides some kind of optimal performance. To say that one system is optimal makes it necessary to define some measure of quality by which to judge the performance of any candidate system. This leads to the requirement for a functional, J, to be minimized, which represents system performance.

Some designers might choose to minimize the maximum amount of error one would expect to see if the true parameter vector, $\mathbf{a}^*$, is anywhere within the admissible subset, $\mathcal{A}$, of the parameter space. One suitable measure is the maximum mean squared error;

$$\begin{aligned}
J^{\infty} &\triangleq \max_{\mathbf{a}\in\mathcal{A}} E\{[\mathbf{x}-\hat{\mathbf{x}}]^{T}\mathbf{W}[\mathbf{x}-\hat{\mathbf{x}}]\} \\
&= \max_{\mathbf{a}\in\mathcal{A}} E\{\mathrm{tr}[\tilde{\mathbf{x}}^{T}\mathbf{W}\tilde{\mathbf{x}}]\} \\
&= \max_{\mathbf{a}\in\mathcal{A}} E\{\mathrm{tr}[\mathbf{W}\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{T}]\} \\
&= \max_{\mathbf{a}\in\mathcal{A}} \mathrm{tr}[\mathbf{W}E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{T}\}] \\
&= \max_{\mathbf{a}\in\mathcal{A}} \mathrm{tr}[\mathbf{W}\boldsymbol{\Psi}] \qquad (8)
\end{aligned}$$

where$E\{\bullet\}$ is the expectation operator and$\boldsymbol{\Psi}$ is the autocorrelation matrix of the estimation error vector, $\tilde{\mathbf{x}}$.

In this case$\mathbf{W}$ is a matrix of weighting coefficients. If a vector of linear combinations of the error, $\mathbf{y} = \mathbf{C}[\mathbf{x}-\hat{\mathbf{x}}]$, is of interest, this functional can be used with $\mathbf{W} = \mathbf{C}^{T}\mathbf{W}'\mathbf{C}$. The example in Chapter III illustrates the use of $\mathbf{W}$ to alter the optimal discretization when the states are of varying interest.

It is easy to see where problems may occur using this functional in a numerical optimization problem. If the maximum error occurred between the i^{th} and j^{th} filter, additional filters could be added anywhere but between these filters without changing the value of the functional. The functional chosen for this research program is the average value of the mean squared estimation error, where the average is taken as the true parameter ranges over the entire admissible parameter set:

$$J^2 \triangleq \frac{\int_{\mathcal{A}} E\{[\mathbf{x}-\hat{\mathbf{x}}]^T \mathbf{W}[\mathbf{x}-\hat{\mathbf{x}}]\}d\mathbf{a}}{\int_{\mathcal{A}} d\mathbf{a}} \tag{9}$$

where

$$\int_{\mathcal{A}} d\mathbf{a} \triangleq \int_{\mathcal{A}_r} \cdots \int_{\mathcal{A}_2} \int_{\mathcal{A}_1} da_1 da_2 \ldots da_r$$

This functional was chosen because it provides information about both the mean and variance of the estimation error across the range of admissible parameter vectors. There are any number of other cost functionals which one might want to minimize. One such cost functional is a linear combination of Equations (8) and (9) to force good overall performance combined with a limit on the maximum error autocorrelation. This is investigated in Section 3.1.5.3 for the example with a cost functional of the form:

$$J = \lambda J^2 + (1-\lambda)J^\infty \tag{10}$$

Besides performance based on estimation error, one might consider performance of the estimator based on convergence to the proper model. One might want to choose the representative parameter set to enhance convergence properties. Measures such as the curvature of an ambiguity function [36] might be maximized to enhance identifiability. There may be some way to use Baram's information measure [4] to formulate a functional which can be used to enhance identifiability. Functionals such as these are not addressed in this dissertation.

In some cases, accurate estimation of the states may not be as important as accurate estimation of the parameter. Figure 3 illustrates one way a parameter estimator might be used. The functional chosen to investigate parameter estimation is:

$$J_a^2 \triangleq \frac{\int_{\mathcal{A}} E\{[\mathbf{a}-\hat{\mathbf{a}}]^T \mathbf{W}[\mathbf{a}-\hat{\mathbf{a}}]\}d\mathbf{a}}{\int_{\mathcal{A}} d\mathbf{a}} \tag{11}$$

It is shown in Chapter III that minimizing this functional does not necessarily produce the same result as minimizing the state estimation functional in Equation (9).

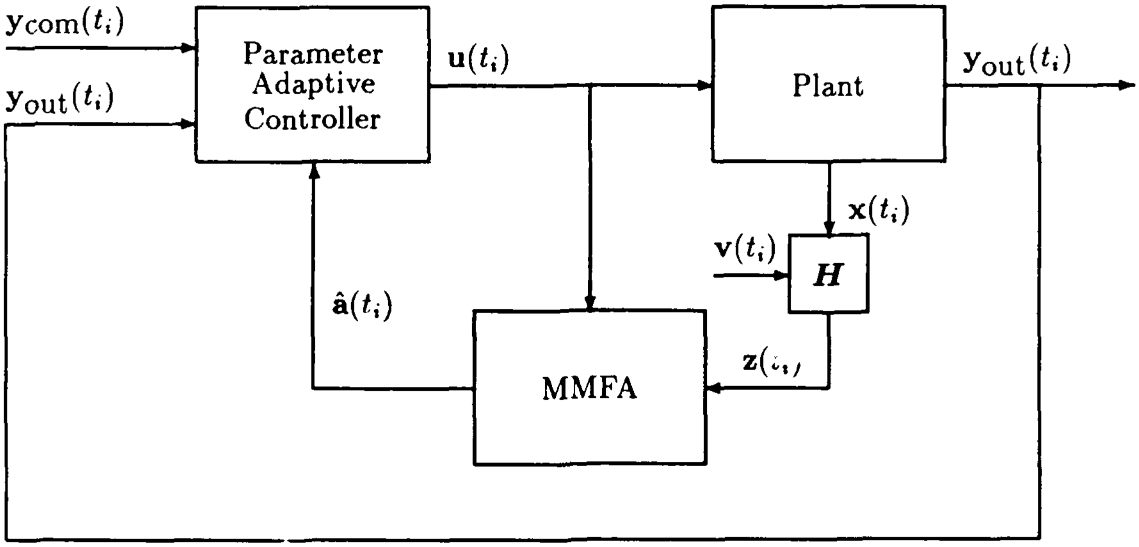


Figure 3. Parameter Estimation in a Control System

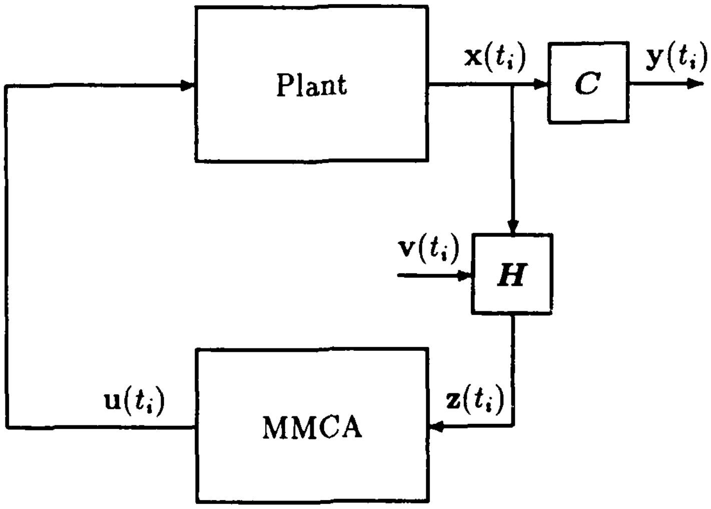


Figure 4. A Multiple Model Adaptive Regulator

When a controller comes into play, it is desirable to minimize the tracking error. Figure 4 is a block diagram of an MMCA used as a regulator where the objective is to control the output vector, $\mathbf{y}(t_i)$, to zero. The cost functional for this system, analogous to Equation (9), can be written as follows:

$$\mathrm{J}_c^2 \triangleq \frac{\int_{\mathcal{A}} \mathrm{E}\{\mathbf{y}^{\mathrm{T}}\mathbf{W}\mathbf{y}\}\mathbf{da}}{\int_{\mathcal{A}} \mathbf{da}} \tag{12}$$

As is the case with the state estimation problem, there are any number of other parameter estimator or controller cost functionals which one may want to minimize, and this research does not preclude using any of them for a similar design effort. The development shown in the sequel may be used to construct analogous design methods.

1.5 General Approach

This section discusses in detail the approach taken to support the hypotheses presented in Section 1.4. Included is a graphical representation of the steps taken in this research to show how the design method was developed and evaluated. This section is meant to provide the reader with a roadmap of the research summarized in this dissertation, and to provide insight about how other similar design methods might be developed.

Hypothesis 1a *There exists a representative subset of the admissible parameter set, which forms the basis of an MMFA, of prespecified size, to provide state estimation with minimum estimation error.*

Approach 1. Start with a single input single output (SISO) linear plant with one parameter ($r = 1$) driven by white Gaussian noise with known values of dynamic driving noise strength (Q) and measurement noise covariance (R).

2. Assume the filters are constant-gain steady-state Kalman filters calculated with the values of Q and R from the truth model.

3. Assume the number of models is specified.

4. Assume the cost criterion is based on steady state error with the estimator converging to the model closest, in the information measure of Baram [4], to the true parameter value (i.e. assume there is enough information in the measurements to force the estimator to converge to the selected filter with probability 1).

5. Derive a relationship between the estimation error of a model and the distance of that model's assumed parameter value from the true parameter value.

6. Develop a procedure for numerically minimizing the cost functional.

7. Evaluate the basic optimization algorithm on a single input single output (SISO) example problem with one parameter and no weighting matrix.

8. Generalize the procedure to account for a vector of r parameters, and to account for multiple input multiple output (MIMO) systems.

9. Evaluate the effect of different weighting matrices on the example problem to exhibit the benefits of this design method.

10. Evaluate the effect of placing a lower bound on the probability to provide for an estimator which can adapt to varying parameters.

11. Investigate other types of estimator mechanizations including a maximum entropy with unit covariance (ME/I) hypothesis [25] in the hypothesis conditional probability calculation. This mechanization forces convergence to the filter with the lowest residual autocorrelation, $\mathrm{E}\{\mathbf{r}^{k\mathsf{T}}(t_i)\mathbf{r}^k(t_i)\}$, instead of the lowest $\mathrm{E}\{\mathbf{r}^{k\mathsf{T}}(t_i)[\boldsymbol{M}^k(t_i)]^{-1}\mathbf{r}^k(t_i)\}$ per Equation (2).

12. Investigate alternative cost functionals including a maximum-error-type cost as shown in Equation (10).

Contribution An automated method for selecting the models in an MMFA.

Notes A graphical representation of this approach is shown in Figure 5 along with the interrelation between this hypothesis and the other hypotheses.

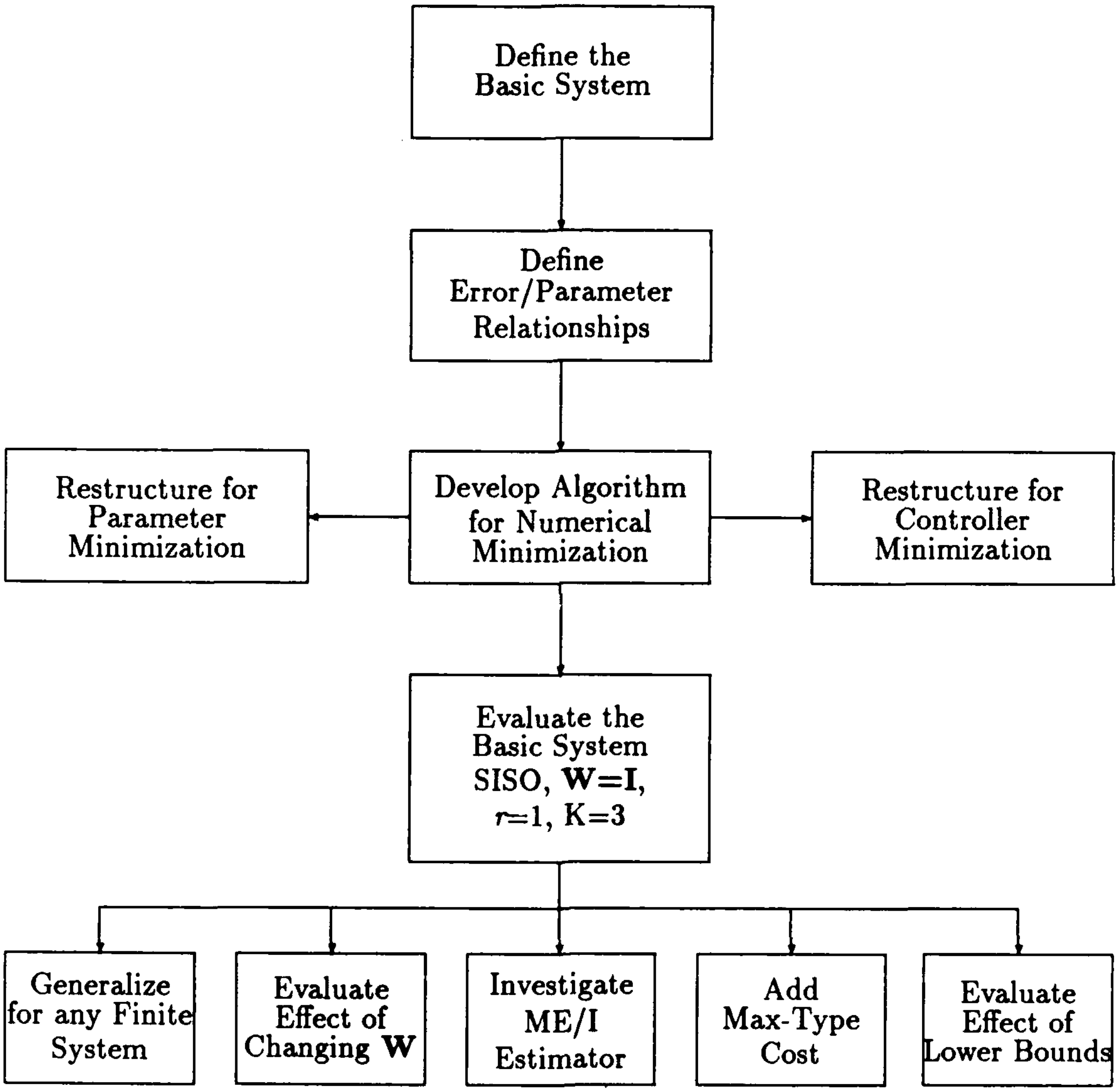


Figure 5. Graphical Representation of Hypothesis 1a

Hypothesis 1b *There is a minimum number of parameters in any representative subset of the admissible parameter set, forming the basis of an MMFA, that provide state estimation with a prespecified bound on the estimation error.*

Approach 1. Use the minimization procedure developed above to define a mapping from the number of models to the expected cost.

2. Choose the minimum number of models from this list which provides at least the level of performance required.

Contribution A method for determining the number of models required for a specified allowable error.

Notes Requires accomplishing the tasks in Hypothesis 1a.

Hypothesis 2a *There exists a representative subset of the admissible parameter set, which forms the basis of an MMFA, of prespecified size, to provide parameter estimation with minimum estimation error.*

Approach 1. Starting with the basic system and algorithm developed for the state estimator, restructure the algorithm to minimize the parameter estimation error.

2. Evaluate the parameter estimation algorithm using the example problem.

3. Evaluate the effect of different weighting matrices on the example problem to exhibit the benefits of this design method.

4. Evaluate the effect of placing a lower bound on the probability to provide for an estimator which can adapt to varying parameters.

5. Investigate other types of estimator mechanizations including a maximum entropy with unit covariance (ME/I) hypothesis [25].

Contribution An automated method for selecting the models in a parameter estimating MMFA.

Notes A graphical representation of this approach is shown in Figure 6.

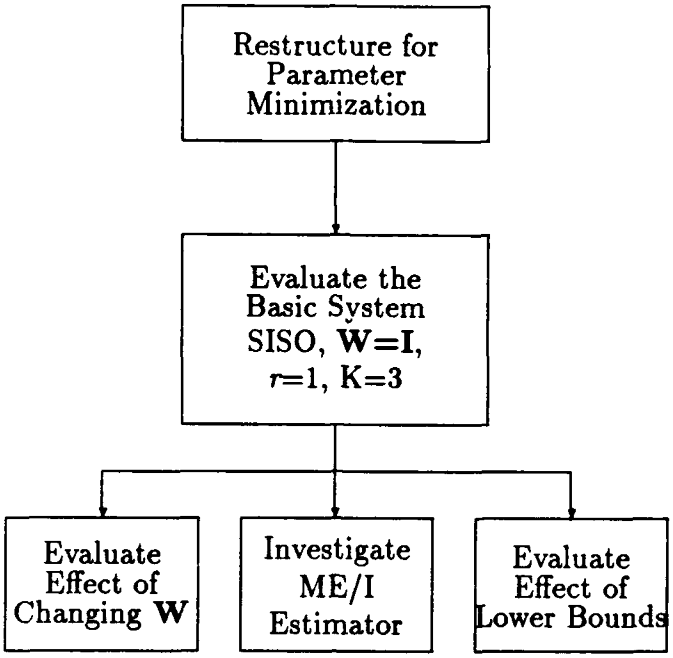

Restructure for Parameter Minimization
Evaluate the Basic System SISO, **W**=**I**, r=1, K=3
Evaluate Effect of Changing **W**
Investigate ME/I Estimator
Evaluate Effect of Lower Bounds

Figure 6. Graphical Representation of Hypothesis 2a

Hypothesis 2b *There is a minimum number of parameters in any representative subset of the admissible parameter set, forming the basis of an MMFA, that provide parameter estimation with a prespecified bound on the estimation error.*

Approach 1. Use the minimization procedure developed above to define a mapping from the number of models to the expected cost.

2. Choose the minimum number of models from this list which provides at least the level of parameter estimation accuracy required.

Contribution A method for finding the number of models required for a specified allowable parameter estimation error.

Notes Requires accomplishing the tasks in Hypothesis 2a.

Hypothesis 3a *There exists a representative subset of the admissible parameter set, which forms the basis of an MMCA, of prespecified size, to provide control with minimum tracking error.*

Approach 1. Starting with the basic system and algorithm developed for the state estimator, restructure the algorithm to minimize the truth state autocorrelation. Define the method of controller design, specifying all necessary design variables. The method of choice is LQG constant gain controllers, because the control gain can be automatically calculated for any given parameter based upon designer specified weighting matrices. Only constant gain filters and controllers are addressed in the sequel.

2. Evaluate the controller optimization algorithm using the example problem.

3. Evaluate the effect of different weighting matrices on the example problem to exhibit the benefits of this design method.

4. Evaluate the effect of placing a lower bound on the probability to provide for an estimator which can adapt to varying parameters.

5. Investigate other types of estimator mechanizations including a maximum entropy with unit covariance (ME/I) hypothesis [25].

Contribution An automated method for selecting the models in an MMCA.

Notes A graphical representation of this approach is shown in Figure 7.

Hypothesis 3b *There is a minimum number of parameters in any representative subset of the admissible parameter set, forming the basis of an MMCA, that provides control with a prespecified bound on the tracking error.*

Approach 1. Use the minimization procedure developed above to define a mapping from the number of models to the expected cost.

2. Choose the minimum number of models from this list which provides at least the level of performance required.

Contribution A method for determining the number of models required for a specified allowable error.

Notes Requires accomplishing the tasks in Hypothesis 2a.

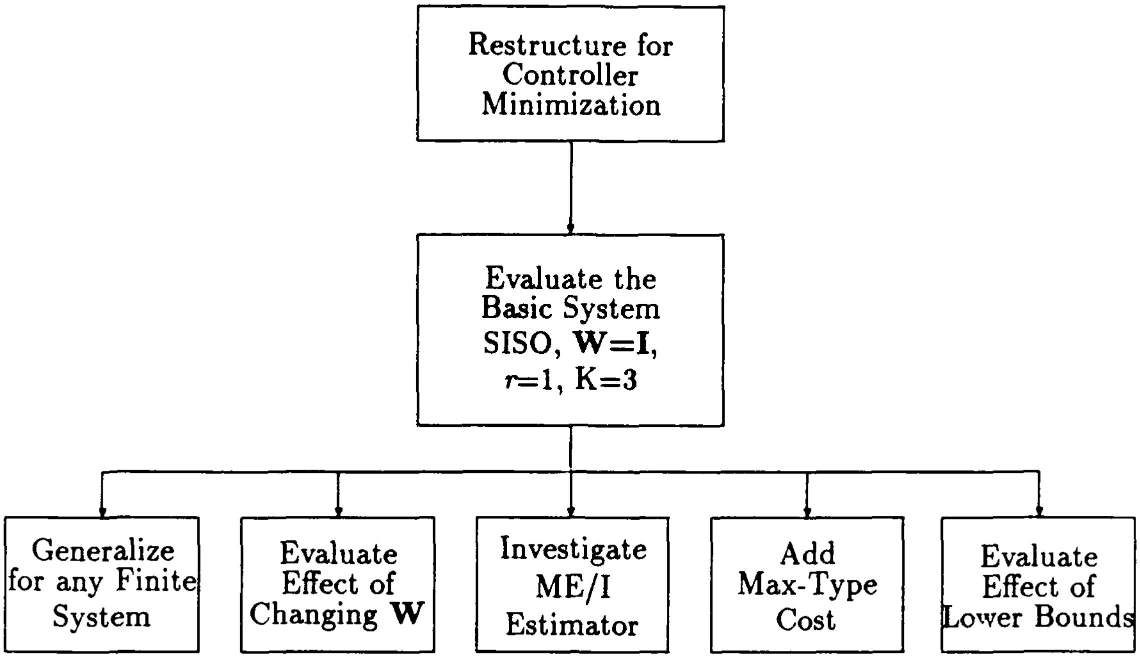

Restructure for Controller Minimization
Evaluate the Basic System SISO, **W**=**I**, r=1, K=3
Generalize for any Finite System
Evaluate Effect of Changing **W**
Investigate ME/I Estimator
Add Max-Type Cost
Evaluate Effect of Lower Bounds

Figure 7. Graphical Representation of Hypothesis 3a

1.6 Assumptions

Some assumptions are necessary to bound the problem and form the basis of the mathematical development of Chapter II. The main assumption is that the structure of the plant is known except for an r-vector of parameters, which is assumed to be a member of an infinite set. It is assumed that this is a bounded and connected set so that it makes sense to numerically integrate over the set in order to find the average mean square error. This is not really necessary, but it gives the problem more physical meaning. It is assumed that there are a finite number of filters available for the estimation, or a finite number of controllers available for the control. The reason is obvious; however, it is also assumed that the designer can specify this number.

The *pseudo*-probabilities will be calculated per Equation (1). The probabilities are referred to as pseudo-probabilities because it is impossible to have an infinite bank of filters, hence, the MMFA might be calculating the probability that each filter is the correct one, when in fact none of them are. There are other methods of calculating a pseudo-probability [25], one of which is discussed in Section 2.1.3.1.

It is assumed throughout that the Bayesian rule, given by Equation (4), is used to calculate the estimate. One last assumption is that the system converges to the model closest, in the information measure of Baram [4], to the true parameter value with probability 1. This inherently assumes the application of persistent excitation, which is a concept well known in the field of adaptive control and discussed in [48, Chapter 6]. Persistent excitation has not been explored in this research.

1.7 Summary

Having been shown to perform well in a number of studies (see Section 1.2), the MMFA and MMCA have been chosen as the basis of this research into adaptive estimation and control. This chapter provides some background material on the structures of these multiple model algorithms.

The MMFA (and MMCA) design problem has been defined as one of choosing a representative parameter set, for the basis of elemental filter (controller) design, that is optimized with regard to some measure of the performance of the resulting system. The performance measure chosen for this research is the mean square error averaged over the admissible parameter set. The remaining chapters follow the outlines presented in Section 1.5 to provide an organized path toward solving the MMFA (and MMCA) design problem.

II. Development

The development of the optimization method is presented in this chapter. The equations describing the variables of interest for state variable estimation, parameter estimation, and state variable control are derived, based on the premise that a finite number of filters will be used.

2.1 Optimizing a Multiple Model State Estimator

As explained in the previous chapter, it is desirable to minimize the cost functional

$$J^2 \triangleq \frac{\int_{\mathcal{A}} E\{[\mathbf{x}-\hat{\mathbf{x}}]^T \mathbf{W}[\mathbf{x}-\hat{\mathbf{x}}]\} d\mathbf{a}}{\int_{\mathcal{A}} d\mathbf{a}} \tag{13}$$

where

$$\int_{\mathcal{A}} d\mathbf{a} \triangleq \int_{\mathcal{A}_r} \cdots \int_{\mathcal{A}_2} \int_{\mathcal{A}_1} da_1 da_2 \ldots da_r$$

Since it is assumed that the admissible parameter set is constant for a given problem, it is necessary only to minimize the numerator of Equation (13). Moreover,

$$\begin{aligned} E\{[\mathbf{x}-\hat{\mathbf{x}}]^T \mathbf{W}[\mathbf{x}-\hat{\mathbf{x}}]\} &= E\{\mathrm{tr}\left[[\mathbf{x}-\hat{\mathbf{x}}]^T \mathbf{W}[\mathbf{x}-\hat{\mathbf{x}}]\right]\} & (14) \\ &= E\{\mathrm{tr}\left[\mathbf{W}[\mathbf{x}-\hat{\mathbf{x}}][\mathbf{x}-\hat{\mathbf{x}}]^T\right]\} & (15) \\ &= \mathrm{tr}[\mathbf{W}\mathbf{\Psi}] & (16) \end{aligned}$$

Thus, one needs to compute the error autocorrelation matrix, $\mathbf{\Psi} = E\{[\mathbf{x}-\hat{\mathbf{x}}][\mathbf{x}-\hat{\mathbf{x}}]^T\}$, as a function of the parameter vector $\mathbf{a}$. This is, in essence, a sensitivity analysis of the effect of parameter variation on each of the filters in the MMFA.

Figure 8 illustrates the concept for a single filter system based on the scalar parameter a^1. The plot shows the value of $\mathrm{tr}[\mathbf{W}\mathbf{\Psi}]$ as a function of the true parameter a^*. The curve represents how the error autocorrelation might increase as the true parameter differs from the parameter used to design the filter. Along with knowing the error autocorrelation for a particular filter, it is necessary to develop the equations

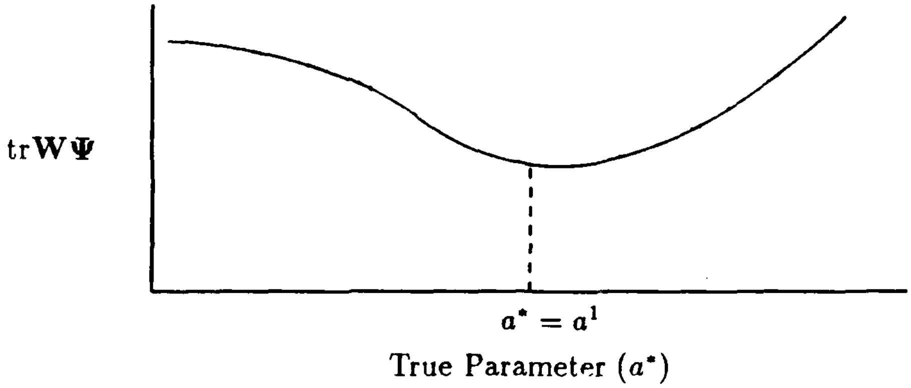


Figure 8. Sensitivity Analysis for a Scalar Parameter

which describe how the estimator converges to a particular filter at a particular point in the parameter space. A numerical integration technique can then be employed to evaluate the functional for a given choice of filters.

2.1.1 Derivation of the Estimator Equations. The equations required to compute the error autocorrelation at each point in parameter space, and the filter selection equations describing how the estimator converges to a particular filter, are presented in following sections.

2.1.1.1 Error Autocorrelation Equations. We want to find out what the true estimation error autocorrelation is when we use a model based on $\mathbf{a}^k$:

$$\mathbf{x}^k(t_{i+1}) = \boldsymbol{\Phi}^k(t_{i+1}, t_i)\mathbf{x}^k(t_i) + \boldsymbol{G}_d^k(t_i)\mathbf{w}_d^k(t_i) \tag{17}$$

$$\mathbf{z}^k(t_i) = \boldsymbol{H}^k(t_i)\mathbf{x}^k(t_i) + \mathbf{v}^k(t_i) \tag{18}$$

which varies from the actual system with the truth model which is based on the true parameter $\mathbf{a}^*$:

$$\mathbf{x}^*(t_{i+1}) = \boldsymbol{\Phi}^*(t_{i+1}, t_i)\mathbf{x}^*(t_i) + \boldsymbol{G}_d^*(t_i)\mathbf{w}_d^*(t_i) \tag{19}$$

$$\mathbf{z}^*(t_i) = \boldsymbol{H}^*(t_i)\mathbf{x}^*(t_i) + \mathbf{v}^*(t_i) \tag{20}$$

and where

$\mathbf{x}^k(t_i)$ is the n^k dimensional state vector in the estimator filter models.

$\mathbf{x}^*(t_i)$ is the n^* dimensional state vector in the truth model; n^* need not be the same as n^k.

$\boldsymbol{\Phi}^k(t_{i+1}, t_i)$ is the $n^k \times n^k$ state transition matrix for the k^{th} filter model.

$\boldsymbol{\Phi}^*(t_{i+1}, t_i)$ is the $n^* \times n^*$ state transition matrix in the truth system model.

$\mathbf{w}_d^k(t_i)$ is an s^k-vector-valued discrete-time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{[\mathbf{w}_d^k(t_i)][\mathbf{w}_d^k(t_j)]^\mathsf{T}\} = \begin{cases} \boldsymbol{Q}_d^k(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases}$$

$\mathbf{w}_d^*(t_i)$ is an s^*-vector-valued discrete-time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{[\mathbf{w}_d^*(t_i)][\mathbf{w}_d^*(t_j)]^\mathsf{T}\} = \begin{cases} \boldsymbol{Q}_d^*(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases}$$

$\boldsymbol{G}_d^k(t_i)$ is the $n^k \times s^k$ noise input matrix for the k^{th} filter model.

$\boldsymbol{G}_d^*(t_i)$ is the $n^* \times s^*$ noise input matrix for the truth model.

$\mathbf{z}^k$ is the m dimensional random measurement vector as modelled in the filters.

$\mathbf{z}^*$ is the m dimensional random vector which represents the "real world" measurement from the truth model.

$\boldsymbol{H}^k(t_i)$ is the $m \times n^k$ measurement matrix as modelled in the filters.

$\boldsymbol{H}^*(t_i)$ is the $m \times n^*$ measurement matrix in the truth system model.

$\mathbf{v}^k(t_i)$ is an m-vector-valued discrete-time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{[\mathbf{v}^k(t_i)][\mathbf{v}^k(t_j)]^\mathsf{T}\} = \begin{cases} \boldsymbol{R}^k(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases}$$

$\mathbf{w}_d^k$ and $\mathbf{v}^k$ are assumed to be independent.

$\mathbf{v}^*(t_i)$ is an m-vector-valued discrete-time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{[\mathbf{v}^*(t_i)][\mathbf{v}^*(t_j)]^{\mathsf{T}}\} = \begin{cases} \boldsymbol{R}^*(t_i) & t_i = t_j \\ 0 & t_i \neq t_j \end{cases}$$

$\mathbf{w}_d^*$ and $\mathbf{v}^*$ are assumed to be independent.

The development uses steady state arguments so the following assumptions are made about time invariance:

$$t_{i+1} - t_i = \text{a constant} \quad \forall i$$

$$\boldsymbol{\Phi}^k(t_{i+1}, t_i) = \boldsymbol{\Phi}^k \quad \forall t$$

$$\boldsymbol{\Phi}^*(t_{i+1}, t_i) = \boldsymbol{\Phi}^* \quad \forall t$$

$$\boldsymbol{H}^k(t_i) = \boldsymbol{H}^k \quad \forall t$$

$$\boldsymbol{H}^*(t_i) = \boldsymbol{H}^* \quad \forall t$$

$$\boldsymbol{G}_d^k(t_i) = \boldsymbol{G}_d^k \quad \forall t$$

$$\boldsymbol{G}_d^*(t_i) = \boldsymbol{G}_d^* \quad \forall t$$

The filters are assumed to be constant-gain filters (i.e., $\boldsymbol{K}^k$ does not vary with time). For the example shown in the sequel, the steady state Kalman filter gain is chosen. It is important to bring out the point that the filters in the MMFA need not be steady state Kalman filters. For this development to apply, it is necessary that the filters have the same structure as a Kalman filter, but any other method of calculating a constant gain that produces a stable estimator (e.g., eigenstructure assignment) may be used. However, to use this method effectively for minimizing the error autocorrelation, an automated method which can determine $\boldsymbol{K}^k$ for any given parameter is suggested. Methods which require hand tuning to determine the gain are *not* recommended.

The steady state Kalman filter gain is used in this dissertation, because the true Kalman filter is the minimum mean squared error estimator for a known system driven by zero mean white Gaussian noise. Since the goal of this research is to minimize the average mean square error, it is natural to use the steady state Kalman filter gain.

The filter selection within the MMFA is determined using the one-step prediction model [4, page 78] as is shown in the next section. For this reason, the one-step prediction model is derived. If $\hat{\mathbf{x}}^k(t_i^-)$ is defined as the state estimate in the k^{th} filter at time t_i before the measurement update, and $\hat{\mathbf{x}}^k(t_i^+)$ as the state estimate in the k^{th} filter at time t_i after the measurement update, we develop a one-step discrete model of the system at t_i^-. Note that the true state vector does not change due to activity in the estimator so $\mathbf{x}^*(t_i^-) = \mathbf{x}^*(t_i^+) = \mathbf{x}^*(t_i)$.

Derive the one-step prediction model from the standard Kalman filter equations [35, page 275] as follows:

$$\hat{\mathbf{x}}^k(t_i^+) = \hat{\mathbf{x}}^k(t_i^-) + \boldsymbol{K}^k[\mathbf{z}^*(t_i) - \boldsymbol{H}^k\hat{\mathbf{x}}^k(t_i^-)] \tag{21}$$

$$\hat{\mathbf{x}}^k(t_{i+1}^-) = \boldsymbol{\Phi}^k\hat{\mathbf{x}}^k(t_i^+) \tag{22}$$

$$\hat{\mathbf{x}}^k(t_{i+1}^-) = \boldsymbol{\Phi}^k\{\hat{\mathbf{x}}^k(t_i^-) + \boldsymbol{K}^k[\boldsymbol{H}^*\mathbf{x}^*(t_i^-) + \mathbf{v}^*(t_i) - \boldsymbol{H}^k\hat{\mathbf{x}}^k(t_i^-)]\} \tag{23}$$

and since we have no measurement update cycle on the true system

$$\mathbf{x}^*(t_{i+1}^-) = \boldsymbol{\Phi}^*\mathbf{x}^*(t_i^-) + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i) \tag{24}$$

At this point we introduce the shorthand notation $\cdot_i^- \triangleq \cdot(t_i^-)$ and can write the one step prediction model as in [4, page 78].

$$\begin{bmatrix} \hat{\mathbf{x}}_{i+1}^{k-} \\ \mathbf{x}_{i+1}^{*-} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^k(\boldsymbol{I} - \boldsymbol{K}^k\boldsymbol{H}^k) & \boldsymbol{\Phi}^k\boldsymbol{K}^k\boldsymbol{H}^* \\ \mathbf{0} & \boldsymbol{\Phi}^* \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_i^{k-} \\ \mathbf{x}_i^{*-} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi}^k\boldsymbol{K}^k \\ \mathbf{0} \end{bmatrix} \mathbf{v}_i^* + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{G}_d^* \end{bmatrix} \mathbf{w}_{di}^* \tag{25}$$

The prediction error equation is now of the form:

$$\begin{bmatrix} \hat{\mathbf{x}}^{k-}_{i+1} \\ \mathbf{x}^{*-}_{i+1} \end{bmatrix} = \boldsymbol{\Upsilon} \begin{bmatrix} \hat{\mathbf{x}}^{k-}_{i} \\ \mathbf{x}^{*-}_{i} \end{bmatrix} + \boldsymbol{G}_o \begin{bmatrix} \mathbf{w}^*_{di} \\ \mathbf{v}^*_{i} \end{bmatrix} \tag{26}$$

where

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \boldsymbol{\Phi}^k(\boldsymbol{I} - \boldsymbol{K}^k\boldsymbol{H}^k) & \boldsymbol{\Phi}^k\boldsymbol{K}^k\boldsymbol{H}^* \\ 0 & \boldsymbol{\Phi}^* \end{bmatrix}$$

$$\boldsymbol{G}_o = \begin{bmatrix} 0 & \boldsymbol{\Phi}^k\boldsymbol{K}^k \\ \boldsymbol{G}^*_d & 0 \end{bmatrix}$$

and $\begin{bmatrix} \mathbf{w}^*_{di} \\ \mathbf{v}^*_{i} \end{bmatrix}$ is an $s^* + m$-vector valued discrete time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{\begin{bmatrix} \mathbf{w}^*_{di} \\ \mathbf{v}^*_{i} \end{bmatrix} [\mathbf{w}^{*\mathrm{T}}_{dj} \mathbf{v}^{*\mathrm{T}}_{j}]\} = \begin{cases} \boldsymbol{Q}_o(t_i) & t_i = t_j \\ 0 & t_i \neq t_j \end{cases}$$

with

$$\boldsymbol{Q}_o(t_i) = \begin{bmatrix} \boldsymbol{Q}^*_d(t_i) & 0 \\ 0 & \boldsymbol{R}^*(t_i) \end{bmatrix}$$

Since $\mathbf{w}^*_d$ and $\mathbf{v}^*$ are assumed to be independent random vectors, they are uncorrelated.

This stucture allows us to calculate the autocorrelation of $\begin{bmatrix} \hat{\mathbf{x}}^{k-}_{i+1} \\ \mathbf{x}^{*-}_{i+1} \end{bmatrix}$ using the matrix equation

$$\mathrm{E}\left\{\begin{bmatrix} \hat{\mathbf{x}}^{k-}_{i+1} \\ \mathbf{x}^{*-}_{i+1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}^{k-\mathrm{T}}_{i+1} & \mathbf{x}^{*-\mathrm{T}}_{i+1} \end{bmatrix}\right\} = \boldsymbol{\Xi}^k_{i+1} = \boldsymbol{\Upsilon}\boldsymbol{\Xi}^k_i\boldsymbol{\Upsilon}^{\mathrm{T}} + \boldsymbol{G}_o\boldsymbol{Q}_o\boldsymbol{G}^{\mathrm{T}}_o \tag{27}$$

If $\boldsymbol{\Upsilon}$ is a contraction, as $i \rightarrow \infty$, $\boldsymbol{\Xi}^k_i$ approaches a constant value which is the steady state prediction error autocorrelation, and is denoted $\boldsymbol{\Xi}^k_{\infty}$.

However, we are not interested in $\begin{bmatrix} \hat{\mathbf{x}}^{k-}_{i+1} \\ \mathbf{x}^{*-}_{i+1} \end{bmatrix}$ but in the estimation error. Since the filter might implement a reduced order model, we must introduce a transformation matrix $\boldsymbol{T}$ which transforms the true state vector into the filter state space by:

$$\mathrm{x}_{\mathrm{implemented}} = \boldsymbol{T}\mathrm{x}_{\mathrm{true}} \tag{28}$$

so let $\tilde{\mathrm{x}}^{k} = \hat{\mathrm{x}}^{k} - \boldsymbol{T}\mathrm{x}^{*}$ denote the estimation error vector.

We want to find the autocorrelation matrix:

$$\mathrm{E}\{[\hat{\mathrm{x}}_i^{k-} - \boldsymbol{T}\mathrm{x}_i^{*-}][\hat{\mathrm{x}}_i^{k-} - \boldsymbol{T}\mathrm{x}_i^{*-}]^{\mathsf{T}}\} = \mathrm{E}\{[\tilde{\mathrm{x}}_i^{k-}][\tilde{\mathrm{x}}_i^{k-}]^{\mathsf{T}}\} \tag{29}$$

Hence, we want to derive

$$\begin{bmatrix} \tilde{\mathrm{x}}_{i+1}^{k-} \\ \mathrm{x}_{i+1}^{*-} \end{bmatrix} = \boldsymbol{\mathcal{L}} \begin{bmatrix} \tilde{\mathrm{x}}_i^{k-} \\ \mathrm{x}_i^{*-} \end{bmatrix} + \boldsymbol{G}_{\mathrm{o}}^{\prime} \begin{bmatrix} \mathrm{w}_{di}^{*} \\ \mathrm{v}_i^{*} \end{bmatrix} \tag{30}$$

to be able to calculate

$$\mathrm{E}\left\{ \begin{bmatrix} \tilde{\mathrm{x}}_{i+1}^{k-} \\ \mathrm{x}_{i+1}^{*-} \end{bmatrix} \begin{bmatrix} \tilde{\mathrm{x}}_{i+1}^{k-\mathsf{T}} & \mathrm{x}_{i+1}^{*-\mathsf{T}} \end{bmatrix} \right\} = \boldsymbol{\Gamma}_{i+1}^{k} = \boldsymbol{\mathcal{L}}\,\boldsymbol{\Gamma}_i^{k}\boldsymbol{\mathcal{L}}^{\mathsf{T}} + \boldsymbol{G}_{\mathrm{o}}^{\prime}\boldsymbol{Q}_{\mathrm{o}}\boldsymbol{G}_{\mathrm{o}}^{\prime\mathsf{T}} \tag{31}$$

We begin by writing the equation for $\tilde{\mathrm{x}}_{i+1}^{k-}$:

$$\begin{aligned} \tilde{\mathrm{x}}_{i+1}^{k-} &= \hat{\mathrm{x}}_{i+1}^{k-} - \boldsymbol{T}\mathrm{x}_{i+1}^{*-} & (32)\\ &= \boldsymbol{\Phi}^{k}\boldsymbol{K}^{k}\boldsymbol{H}^{*}\mathrm{x}_i^{*-} + \boldsymbol{\Phi}^{k}(\boldsymbol{I} - \boldsymbol{K}^{k}\boldsymbol{H}^{k})\hat{\mathrm{x}}_i^{k-} - \boldsymbol{T}\boldsymbol{\Phi}^{*}\mathrm{x}_i^{*-} + \boldsymbol{\Phi}^{k}\boldsymbol{K}^{k}\mathrm{v}_i^{*} \\ &\qquad - \boldsymbol{T}\boldsymbol{G}_d^{*}\mathrm{w}_{di}^{*} & (33)\\ &= \boldsymbol{\Phi}^{k}\boldsymbol{K}^{k}\boldsymbol{H}^{*}\mathrm{x}_i^{*-} + \boldsymbol{\Phi}^{k}\hat{\mathrm{x}}_i^{k-} - \boldsymbol{\Phi}^{k}\boldsymbol{K}^{k}\boldsymbol{H}^{k}\hat{\mathrm{x}}_i^{k-} - \boldsymbol{T}\boldsymbol{\Phi}^{*}\mathrm{x}_i^{*-} + \boldsymbol{\Phi}^{k}\boldsymbol{K}^{k}\mathrm{v}_i^{*} \\ &\qquad - \boldsymbol{T}\boldsymbol{G}_d^{*}\mathrm{w}_{di}^{*} & (34) \end{aligned}$$

Now define

$$\boldsymbol{T}\boldsymbol{\Delta\Phi} \triangleq \boldsymbol{\Phi}^{k}\boldsymbol{T} - \boldsymbol{T}\boldsymbol{\Phi}^{*} \tag{35}$$

and

$$\boldsymbol{\Delta H} \triangleq \boldsymbol{H}^{k}\boldsymbol{T} - \boldsymbol{H}^{*} \tag{36}$$

Rearranging these equations gives

$$\boldsymbol{T}\boldsymbol{\Phi}^{*} = \boldsymbol{\Phi}^{k}\boldsymbol{T} - \boldsymbol{T}\boldsymbol{\Delta\Phi} \tag{37}$$

$$\boldsymbol{H}^{*} = \boldsymbol{H}^{k}\boldsymbol{T} - \boldsymbol{\Delta H} \tag{38}$$

Substitute Equations (37) and (38) into Equation (34) to get

$$\begin{aligned}\tilde{\mathbf{x}}_{i+1}^{k-} &= \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{H}^k \boldsymbol{T} \mathbf{x}_i^{*-} - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{\Delta H} \mathbf{x}_i^{*-} + \boldsymbol{\Phi}^k \hat{\mathbf{x}}_i^{k-} - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{H}^k \hat{\mathbf{x}}_i^{k-} \\ &\quad - \boldsymbol{\Phi}^k \boldsymbol{T} \mathbf{x}_i^{*-} + \boldsymbol{T \Delta \Phi} \mathbf{x}_i^{*-} + \boldsymbol{\Phi}^k \boldsymbol{K}^k \mathbf{v}_i^* - \boldsymbol{T G}_d^* \mathbf{w}_{di}^* \qquad (39)\end{aligned}$$

Rearrange and collect terms to obtain

$$\begin{aligned}\tilde{\mathbf{x}}_{i+1}^{k-} &= \boldsymbol{\Phi}^k \hat{\mathbf{x}}_i^{k-} - \boldsymbol{\Phi}^k \boldsymbol{T} \mathbf{x}_i^{*-} - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{H}^k \hat{\mathbf{x}}_i^{k-} + \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{H}^k \boldsymbol{T} \mathbf{x}_i^{*-} + \boldsymbol{T \Delta \Phi} \mathbf{x}_i^{*-} \\ &\quad - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{\Delta H} \mathbf{x}_i^{*-} + \boldsymbol{\Phi}^k \boldsymbol{K}^k \mathbf{v}_i^* - \boldsymbol{T G}_d^* \mathbf{w}_{di}^* \qquad (40)\end{aligned}$$

$$\begin{aligned}\tilde{\mathbf{x}}_{i+1}^{k-} &= \boldsymbol{\Phi}^k (\hat{\mathbf{x}}_i^{k-} - \boldsymbol{T} \mathbf{x}_i^{*-}) - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{H}^k (\hat{\mathbf{x}}_i^{k-} - \boldsymbol{T} \mathbf{x}_i^{*-}) + \boldsymbol{T \Delta \Phi} \mathbf{x}_i^{*-} \\ &\quad - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{\Delta H} \mathbf{x}_i^{*-} + \boldsymbol{\Phi}^k \boldsymbol{K}^k \mathbf{v}_i^* - \boldsymbol{T G}_d^* \mathbf{w}_{di}^* \qquad (41)\end{aligned}$$

$$\begin{aligned}\tilde{\mathbf{x}}_{i+1}^{k-} &= \boldsymbol{\Phi}^k \tilde{\mathbf{x}}_i^{k-} - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{H}^k \tilde{\mathbf{x}}_i^{k-} + \boldsymbol{T \Delta \Phi} \mathbf{x}_i^{*-} - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{\Delta H} \mathbf{x}_i^{*-} \\ &\quad + \boldsymbol{\Phi}^k \boldsymbol{K}^k \mathbf{v}_i^* - \boldsymbol{T G}_d^* \mathbf{w}_{di}^* \qquad (42)\end{aligned}$$

$$\begin{aligned}\tilde{\mathbf{x}}_{i+1}^{k-} &= (\boldsymbol{\Phi}^k - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{H}^k) \tilde{\mathbf{x}}_i^{k-} + (\boldsymbol{T \Delta \Phi} - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{\Delta H}) \mathbf{x}_i^{*-} \\ &\quad + \boldsymbol{\Phi}^k \boldsymbol{K}^k \mathbf{v}_i^* - \boldsymbol{T G}_d^* \mathbf{w}_{di}^* \qquad (43)\end{aligned}$$

and as in the previous development

$$\mathbf{x}(t_{i+1}^{*-}) = \boldsymbol{\Phi}^*(t_{i+1}, t_i) \mathbf{x}(t_i^{*-}) + \boldsymbol{G}_d^* \mathbf{w}_{di}^* \qquad (44)$$

We can now write the one-step prediction error

$$\begin{aligned}\begin{bmatrix} \tilde{\mathbf{x}}_{i+1}^{k-} \\ \mathbf{x}_{i+1}^{*-} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\Phi}^k (\boldsymbol{I} - \boldsymbol{K}^k \boldsymbol{H}^k) & \boldsymbol{T \Delta \Phi} - \boldsymbol{\Phi}^k \boldsymbol{K}^k \boldsymbol{\Delta H} \\ \mathbf{0} & \boldsymbol{\Phi}^* \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_i^{k-} \\ \mathbf{x}_i^{*-} \end{bmatrix} \\ &\quad + \begin{bmatrix} -\boldsymbol{T G}_d^* & \boldsymbol{\Phi}^k \boldsymbol{K}^k \\ \boldsymbol{G}_d^* & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{di}^* \\ \mathbf{v}_i^* \end{bmatrix} \qquad (45)\end{aligned}$$

which is of the form

$$\begin{bmatrix} \tilde{\mathbf{x}}^{k-}_{i+1} \\ \mathbf{x}^{*-}_{i+1} \end{bmatrix} = \boldsymbol{\mathcal{L}} \begin{bmatrix} \tilde{\mathbf{x}}^{k-}_{i} \\ \mathbf{x}^{*-}_{i} \end{bmatrix} + G^{'}_{o} \begin{bmatrix} \mathbf{w}^{*}_{di} \\ \mathbf{v}^{*}_{i} \end{bmatrix} \tag{46}$$

Thus, we can calculate

$$\mathrm{E}\left\{ \begin{bmatrix} \tilde{\mathbf{x}}^{k-}_{i+1} \\ \mathbf{x}^{*-}_{i+1} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{k-\mathsf{T}}_{i+1} & \mathbf{x}^{*-\mathsf{T}}_{i+1} \end{bmatrix} \right\} = \boldsymbol{\Gamma}^{k}_{i+1} = \boldsymbol{\mathcal{L}}\boldsymbol{\Gamma}^{k}_{i}\boldsymbol{\mathcal{L}}^{\mathsf{T}} + G^{'}_{o}Q_{o}G^{'\mathsf{T}}_{o} \tag{47}$$

If $\boldsymbol{\mathcal{L}}$ is a contraction, as $i \to \infty$, $\boldsymbol{\Gamma}^{k}_{i}$ approaches a constant value which is the steady state prediction error autocorrelation, and is denoted $\boldsymbol{\Gamma}^{k}_{\infty}$.

The upper left partition of $\boldsymbol{\Gamma}^{k}_{i}$, $\mathrm{E}\{[\tilde{\mathbf{x}}^{k-}_{i}][\tilde{\mathbf{x}}^{k-}_{i}]^{\mathsf{T}}\}$, is $\boldsymbol{\Psi}^{k-}$, the autocorrelation of the estimator prediction error in the k^{th} filter. We calculate $\mathrm{E}\{[\tilde{\mathbf{x}}^{k-}_{i}]^{\mathsf{T}}\mathbf{W}[\tilde{\mathbf{x}}^{k-}_{i}]\}$ by:

$$\begin{aligned} \mathrm{E}\{[\tilde{\mathbf{x}}^{k-}_{i}]^{\mathsf{T}}\mathbf{W}[\tilde{\mathbf{x}}^{k-}_{i}]\} &= \mathrm{E}\{\mathrm{tr}[[\tilde{\mathbf{x}}^{k-}_{i}]^{\mathsf{T}}\mathbf{W}[\tilde{\mathbf{x}}^{k-}_{i}]]\} \\ &= \mathrm{E}\{\mathrm{tr}[\mathbf{W}[\tilde{\mathbf{x}}^{k-}_{i}][\tilde{\mathbf{x}}^{k-}_{i}]^{\mathsf{T}}]\} \\ &= \mathrm{tr}[\mathbf{W}\mathrm{E}\{[\tilde{\mathbf{x}}^{k-}_{i}][\tilde{\mathbf{x}}^{k-}_{i}]^{\mathsf{T}}\}] \\ &= \mathrm{tr}[\mathbf{W}\boldsymbol{\Psi}^{k-}] \end{aligned} \tag{48}$$

The complete cost functional can be computed as

$$\mathrm{J}^{2} \triangleq \frac{\int_{\mathcal{A}} \mathrm{tr}[\mathbf{W}\boldsymbol{\Psi}^{sel-}]\mathrm{d}\mathbf{a}}{\int_{\mathcal{A}} \mathrm{d}\mathbf{a}} \tag{49}$$

where *sel* denotes the filter selected at any particular value of the parameter vector. We assume that the estimator converges to one particular filter so that $p^{sel} = 1$, and all other p^{k} are zero. The equations governing which filter is selected are developed in the next section. In practice, a lower bound is put on the hypothesis conditional probability for each filter. This prevents the probability of any filter from converging to zero, and allows the estimator to adapt to a varying parameter. In this case, the true estimator error autocorrelation for a Bayesian MMFA would be a probabilistically weighted average of the individual error autocorrelations calculated in Equation (48). This is explored in Sections 2.1.3.3 and 3.1.5.4.

2.1.1.2 Equations Governing Filter Selection Within the Estimator. In his PhD dissertation [4], Baram presents the conditions for convergence within the MMFA. If sufficient conditions exist for the MMFA to converge, it will converge to the filter based on the parameter closest to the true parameter with respect to Baram's information metric. Baram's development makes it necessary to define some terms before stating the conditions for filter selection.

$\boldsymbol{M}^k$ is the steady state prediction error covariance of the residuals calculated within the k^{th} filter by

$$\boldsymbol{M}^k \triangleq [\boldsymbol{H}^k][\boldsymbol{P}^k]^-[\boldsymbol{H}^k]^\mathsf{T} + \boldsymbol{R}^k \tag{50}$$

where $[\boldsymbol{P}^k]^-$ is the prediction error covariance of the states as calculated in the k^{th} filter.

Since the models for the filters do not match the truth model, this value calculated within the filter is not the true prediction error covariance. Assuming the truth model of Equations (19) and (20) is accurate and the filters are based on Equations (17) and (18),

$\boldsymbol{N}^k$ is the actual steady state prediction error autocorrelation in the k^{th} filter generated by [4, Eqn. 5.9]

$$\begin{aligned}
\boldsymbol{N}^k &\triangleq [-\boldsymbol{H}^k\ \boldsymbol{H}^*]\boldsymbol{\Xi}^k_\infty[-\boldsymbol{H}^k\ \boldsymbol{H}^*]^\mathsf{T} + \boldsymbol{R}^* && (51)\\
&= [-\boldsymbol{H}^k\ -\boldsymbol{\Delta H}]\boldsymbol{\Gamma}^k_\infty[-\boldsymbol{H}^k\ -\boldsymbol{\Delta H}]^\mathsf{T} + \boldsymbol{R}^* && (52)\\
&= [\boldsymbol{H}^k\ \boldsymbol{\Delta H}]\boldsymbol{\Gamma}^k_\infty[\boldsymbol{H}^k\ \boldsymbol{\Delta H}]^\mathsf{T} + \boldsymbol{R}^* && (53)
\end{aligned}$$

where $\boldsymbol{\Xi}^k_\infty$ is the steady state value calculated in Equation (27) and $\boldsymbol{\Gamma}^k_\infty$ is the steady state value calculated in Equation (47).

ℓ^k is defined as the proximity of the k^{th} filter generated by [4, Eqn. 5.16]

$$\ell^k \triangleq \log_e \mid \boldsymbol{M}^k \mid + \mathrm{tr}[[\boldsymbol{M}^k]^{-1}\boldsymbol{N}^k] \tag{54}$$

If the conditions exist for the MMFA to converge [4, Theorem 5.2], it will converge to the k^{th} filter governed by

$$\ell^k = \min \ell^j \qquad j = 1, \ldots, \mathrm{K} \tag{55}$$

2.1.2 Estimator Design Algorithm. The following is a procedure by which the cost functional of Equation (13) can be numerically approximated and minimized

1.) Start by describing the system in terms of the parameter vector **a**, specifying the structure of the truth system and the filters to be implemented in the estimator.

2.) Choose K, the number of filters to be implemented in the estimator.

3.) Choose a representative parameter set to begin the minimization. A reasonable choice might be made by dividing the admissible parameter region into equal intervals and choosing the midpoint of each interval, though each problem may require a different starting point. Insight as to what represents a reasonable choice can be gained by plotting the optimal estimation error autocorrelation. That is, plot Equation (48) for a coarse discretization of the parameter space assuming $\mathbf{a}^k = \mathbf{a}^*$ at each point. The larger tr$[\mathbf{W}\mathbf{\Psi}^-]$ is, the closer the representative parameters should be.

4.) Choose a numerical integration technique which will be used to evaluate Equation (13). This integration provides the value of the functional for any specific choice of representative parameter set. Equation (49) defines the curve as a function of the parameter, and is used to evaluate the curve at any required point in the interval. The information required to set up Equations (45) and (47) is chosen based on the proper filter selection per Equation (55).

5.) Step 4 generates a numerical approximation to the value J(**a**) for a prespecified choice of parameter vector **a**. The problem is now a vector minimization problem. Choose a numerical vector minimization technique and proceed to minimize J(**a**) using the procedure in step 4 to evaluate the functional for each parameter vector.

Step 4 is computationally intensive because an algebraic Riccati equation is solved at each point evaluated on the curve. Step 5 is also computationally intensive because derivative information is not explicitly available, and must be computed with difference approximations. The numerical integration and minimization techniques must be chosen with this in mind. The extended trapezoidal rule [51, Eqn. (4.1.11)] is the numerical integration technique used in the example of Chapter III. This technique is a reasonable choice because it provides sufficient accuracy for the example problem and involves minimal computational loading. An empirical comparison of several techniques (including the forward rectangular rule, and alternative extended Simpson's rule [51]) on the example problem resulted in negligible difference (.00012 average relative change) in the value of the cost functional. The extended trapezoidal rule also has the desirable characteristic of corresponding closely to the method used to analyze simulation results, where the mean square error is averaged over the number of simulations.

The vector minimization is done using a Broyden-Fletcher-Goldfarb-Shanno (BFGS) variable metric method for unconstrained minimization using one-sided difference approximations in the software package ADS [62]. This method was chosen because it is efficient and supposedly requires fewer evaluations than the Davidon-Fletcher-Powell algorithm to achieve the same level of accuracy [11]. In practice, (for this research and in [63, page 97]) there is no discernible difference between these two methods. The techniques chosen produced good results for the example of Chapter III.

2.1.3 Modifications to the Basic Estimator Optimization Algorithm. Recall, from Section 1.5 and specifically Figure 5, that there are some modifications to the basic design algorithm which are desirable from a practical standpoint. The algorithm as presented is general enough to account for MIMO systems and parameter vectors. It also incorporates a weighting matrix, $\mathbf{W}$, to allow the designer to perform tradeoffs between states. The effect of this weighting matrix is explored in Section 3.1.5.1. The following sections explore the other extensions of the basic problem which require modifications to the basic optimization algorithm.

2.1.3.1 The Maximum Entropy with Identity Covariance (ME/I) Hypothesis. The MMFA described in Section 1.2.1 was developed based on the assumption that one of the filters in the bank is based upon an accurate model of the system. The probabilities calculated by Equation (1) would be true probabilities if this was so. The residuals of the correct filter would follow a Gaussian distribution with covariance $\boldsymbol{M}^k$, given by Equation (3). A Gaussian distribution is the maximum entropy distribution for the given covariance $\boldsymbol{M}^k$, so a filter based on this hypothesis is called an ME/M filter [25].

The measurement conditional density function for the k^{th} filter is calculated as follows:

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}^k, \mathbf{Z}_{i-1}) = \frac{1}{(2\pi)^{m/2} \mid \boldsymbol{M}^k(t_i) \mid^{1/2}} \exp\{-\frac{1}{2}\mathbf{r}^{k\mathsf{T}}(t_i)[\boldsymbol{M}^k(t_i)]^{-1}\mathbf{r}^k(t_i)\} \tag{56}$$

Since we are approximating a system having an uncountable infinity of possible parameter values, with an MMFA consisting of a finite number of filters, the correct model is almost surely different from any filter in the MMFA. Hence, the probabilities calculated are *pseudo*-probabilities, and the residuals almost surely have a covariance different from any $\boldsymbol{M}^k$.

If we assume instead that the residuals are Gaussian with a covariance matrix equal to the identity matrix, we are assuming the residuals follow a "maximally noncommittal residual distribution" [25, page 26]. Making this assumption would lead us to use a Maximum Entropy with Identity covariance or ME/I filter. The measurement conditional density function for the k^{th} filter is then calculated as follows:

$$f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}^k, \mathbf{Z}_{i-1}) = \frac{1}{(2\pi)^{m/2} \mid \boldsymbol{I} \mid^{1/2}} \exp\{-\frac{1}{2}\mathbf{r}^{k\mathsf{T}}(t_i)\boldsymbol{I}\mathbf{r}^k(t_i)\} \tag{57}$$

The MMFA selects the filter with the minimum proximity given by

$$\ell^k = \log_e \mid \boldsymbol{I} \mid + \mathrm{tr}[\boldsymbol{I}^{-1}\boldsymbol{N}^k] = \mathrm{tr}[\boldsymbol{N}^k] \tag{58}$$

where $\boldsymbol{N}^k$ is given by Equation (53). Thus the filter with the lowest residual autocorrelation is chosen.

We can use this property of the ME/I filter to our advantage. Since the hypothesis conditional probabilities are calculated apart from the individual filters, we can use the ME/I hypothesis to update the probabilities without changing the gain in the estimating filters. In other words, the calculations internal to each elemental filter assume that the covariance of the residuals in that filter are well modelled as $\boldsymbol{M}^k(t_i)$, whereas the decision logic external to each elemental filter assumes instead that this covariance is the identity matrix. Thus, an MMFA using the ME/I hypothesis selects the ME/M filter with the lowest residual error autocorrelation. By replacing Equation (54) with Equation (58), the optimization algorithm developed in Section 2.1.2 can be u^sed to optimize such a system.

2.1.3.2 Alternative Cost Functionals. It is assumed, in the development of Section 2.1, that the design objective is to minimize the *average* mean square estimation error. Many system specifications include upper bounds on the acceptable error. To address these specifications, a maximum-type cost functional is appropriate. Such a cost functional is the maximum mean squared error

$$\mathrm{J}^{\infty} = \max_{\mathbf{a}\in\mathcal{A}} \mathrm{tr}[\mathbf{W}\boldsymbol{\Psi}] \tag{59}$$

It is easy to see where problems may occur using this functional in a numerical optimization problem. The maximum error autocorrelation is likely to occur between two filters, or between a filter and a boundary. This would allow us to move the other filter parameters without affecting the cost, making the derivative of the cost in a gradient optimization algorithm equal to zero in the those directions. The optimization algorithm would assume that the cost functional was minimized in those directions, and ignore the corresponding values. Hence, numerical optimization routines would experience problems due to the flatness of the cost functional surface over the parameter space.

To preclude this minimization problem, we propose the hybrid cost functional

$$\mathrm{J} = \lambda \mathrm{J}^2 + (1-\lambda)\mathrm{J}^{\infty} \tag{60}$$

where λ is allowed to vary between a small ϵ and 1. This cost functional maintains the characteristics of the basic cost functional while placing some weight on the maximum prediction error autocorrelation. It is a simple matter to keep track of the maximum error autocorrelation found during the calculation of J^2, then compute J from Equation (60). This cost functional is explored in Section 3.1.5.3, and is shown to provide an MMFA with lower *maximum* mean square error than the basic algorithm at the cost of a higher *average* mean square error.

2.1.3.3 Avoiding Estimator Lockup. Recall that the conditional densities are used to update the probabilities at each sample instant by the calculation [53]:

$$p^k(t_i) = \frac{f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}^k, \mathbf{Z}_{i-1})p^k(t_{i-1})}{\sum_{j=1}^{K} f_{\mathbf{z}(t_i)|\mathbf{a},\mathbf{Z}(t_{i-1})}(\mathbf{z}_i \mid \mathbf{a}^j, \mathbf{Z}_{i-1})p^j(t_{i-1})} \tag{61}$$

Baram proved [4, Thm.5.2] that the MMFA will converge to the filter selected by Equation (54). This is the basis for assuming that the probability of the selected filter is one. Once the probability of the selected filter reaches one, the other probabilities are zero and the probability calculation will experience lockup. One can see from Equation (61) that if $p^k(t_{i-1}) = 0$, then $p^k(t_i) = 0$.

In practice, the designer will select a lower bound for the probabilities to keep the system open for adaptation if the true parameter changes. Let $p_{\min}$ denote the lower bound on the probabilities. Whereas it is assumed, in the development of Section 2.1.1.1, that the estimator has converged and the selected filter has a probability equal to one, the state estimate actually converges to

$$\begin{aligned} \hat{\mathbf{x}}(t_i^+) &= \sum_{k=1}^{K} \hat{\mathbf{x}}^k(t_i^+)p^k(t_i) \qquad (62) \\ &= \{[1 - (K \times p_{\min})]\hat{\mathbf{x}}^{sel}(t_i^+)\} + \sum_{k=1}^{K} \hat{\mathbf{x}}^k(t_i^+)p_{\min} \qquad (63) \end{aligned}$$

where *sel* denotes the selected filter per Equation (55).

The calculations must be modified to take this into account. If we construct the augmented state vector,

$$\mathbf{x}_a(t_i^-) \triangleq \begin{bmatrix} \hat{\mathbf{x}}^1(t_i^-) \\ \vdots \\ \hat{\mathbf{x}}^K(t_i^-) \\ \mathbf{x}^*(t_i^-) \end{bmatrix} \tag{64}$$

the estimation error can be defined as

$$\begin{aligned} \tilde{\mathbf{x}}(t_i^-) &= \hat{\mathbf{x}}(t_i^-) - \boldsymbol{T}\mathbf{x}^*(t_i^-) & (65) \\ &= \sum_{k=1}^{K} p^k(t_{i-1})\hat{\mathbf{x}}^k(t_i^-) - \boldsymbol{T}\mathbf{x}^*(t_i^-) & (66) \\ &= [p^1\mathbf{I}_{n^1} \cdots p^k\mathbf{I}_{n^k} \; -\boldsymbol{T}]\mathbf{x}_a(t_i^-) & (67) \end{aligned}$$

Defining the probability matrix as

$$\mathbf{A}_p \triangleq \left[p^1\mathbf{I}_{n^1} \; \cdots \; p^k\mathbf{I}_{n^k} \; -\boldsymbol{T}\right] \tag{68}$$

the value of $\mathrm{E}\{[\tilde{\mathbf{x}}^-]^{\mathsf{T}}\mathbf{W}[\tilde{\mathbf{x}}^-]\}$ can be calculated by

$$\begin{aligned} \mathrm{E}\{[\tilde{\mathbf{x}}^-]^{\mathsf{T}}\mathbf{W}[\tilde{\mathbf{x}}^-]\} &= \mathrm{E}\{\mathrm{tr}[[\tilde{\mathbf{x}}^-]^{\mathsf{T}}\mathbf{W}[\tilde{\mathbf{x}}^-]]\} \\ &= \mathrm{E}\{\mathrm{tr}[\mathbf{W}[\tilde{\mathbf{x}}^-][\tilde{\mathbf{x}}^-]^{\mathsf{T}}]\} \\ &= \mathrm{E}\{\mathrm{tr}[\mathbf{W}[\mathbf{A}_p\mathbf{x}_a^-][\mathbf{A}_p\mathbf{x}_a^-]^{\mathsf{T}}]\} \\ &= \mathrm{E}\{\mathrm{tr}[\mathbf{W}[\mathbf{A}_p\mathbf{x}_a^-][\mathbf{x}_a^-]^{\mathsf{T}}\mathbf{A}_p^{\mathsf{T}}]\} \\ &= \mathrm{tr}[\mathbf{W}\mathbf{A}_p\mathrm{E}\{[\mathbf{x}_a^-][\mathbf{x}_a^-]^{\mathsf{T}}\}\mathbf{A}_p^{\mathsf{T}}] & (69) \end{aligned}$$

To calculate the augmented autocorrelation matrix, $\mathrm{E}\{[\mathbf{x}_a^-][\mathbf{x}_a^-]^{\mathsf{T}}\} = \boldsymbol{\Psi}_a$, construct the augmented system, from Equation (25), as follows:

$$\mathbf{x}^-_{a\,i+1} = \begin{bmatrix} \boldsymbol{\Phi}^1(\boldsymbol{I}-\boldsymbol{K}^1\boldsymbol{H}^1) & \cdots & 0 & \boldsymbol{\Phi}^1\boldsymbol{K}^1\boldsymbol{H}^* \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \boldsymbol{\Phi}^k(\boldsymbol{I}-\boldsymbol{K}^k\boldsymbol{H}^k) & \boldsymbol{\Phi}^k\boldsymbol{K}^k\boldsymbol{H}^* \\ 0 & \cdots & 0 & \boldsymbol{\Phi}^* \end{bmatrix} \mathbf{x}^-_{a\,i}$$

$$+ \begin{bmatrix} 0 & \boldsymbol{\Phi}^1\boldsymbol{K}^1 \\ \vdots & \vdots \\ 0 & \boldsymbol{\Phi}^k\boldsymbol{K}^k \\ \boldsymbol{G}^*_d & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}^*_{di} \\ \mathbf{v}^*_i \end{bmatrix} \qquad (70)$$

or

$$\mathbf{x}^-_{a\,i+1} = \boldsymbol{\Upsilon}_a\, \mathbf{x}^-_{a\,i} + \boldsymbol{G}_a \begin{bmatrix} \mathbf{w}^*_{di} \\ \mathbf{v}^*_i \end{bmatrix} \qquad (71)$$

The steady state autocorrelation of the augmented state vector, $\boldsymbol{\Psi}_{a\infty}$, is calculated by letting

$$\boldsymbol{\Psi}_{a\,i+1} = \boldsymbol{\Upsilon}_a \boldsymbol{\Psi}_{a\,i} \boldsymbol{\Upsilon}_a^{\mathsf{T}} + \boldsymbol{G}_a \boldsymbol{Q}_o \boldsymbol{G}_a^{\mathsf{T}} \qquad (72)$$

go to steady state conditions, with $\boldsymbol{Q}_o$ from Equation (27). Hence,

$$\mathrm{E}\{[\tilde{\mathbf{x}}^-]^{\mathsf{T}}\mathbf{W}[\tilde{\mathbf{x}}^-]\} = \mathrm{tr}[\mathbf{W}\mathbf{A}_p\boldsymbol{\Psi}_{a\infty}\mathbf{A}_p^{\mathsf{T}}] \qquad (73)$$

Using this in place of Equation (48) allows us to calculate the cost of a system for any given lower bound on the probability.

2.2 Optimizing for Parameter Estimation

One of the desired estimator structures discussed in Chapter I is that of a system where an estimate of the parameter is needed without the requirement for a good state estimate. Figure 9 (which is a repeat of Figure 3) illustrates such a system. A procedure analogous to that of Section 2.1.2 is used to find a representative parameter set which provides the best parameter estimation.

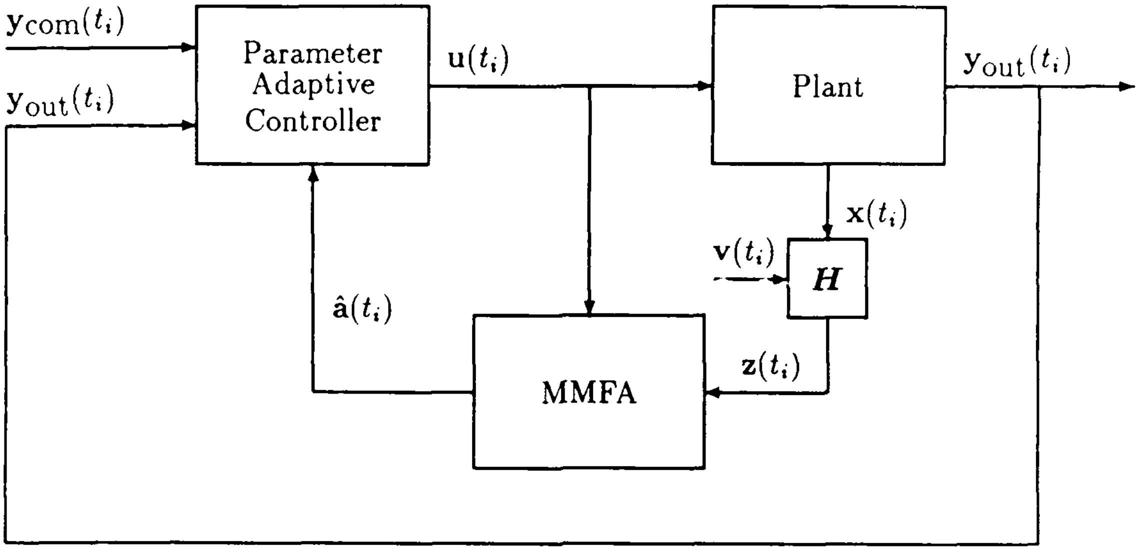


Figure 9. Parameter Estimator Structure

2.2.1 Parameter Estimator Design Algorithm. For the best parameter estimation performance, it is desirable to minimize the cost functional

$$J_a^2 \triangleq \frac{\int_{\mathcal{A}} E\{[\mathbf{a}-\hat{\mathbf{a}}]^T \mathbf{W}[\mathbf{a}-\hat{\mathbf{a}}]\} d\mathbf{a}}{\int_{\mathcal{A}} d\mathbf{a}} \tag{74}$$

In analogy with the discussion below Equation (49), it is assumed that the estimator converges to one particular filter so that $p^{sel} = 1$, and all other p^k are zero, where sel denotes the filter selected at any particular value of the parameter vector. The equations governing which filter is selected are developed in Section 2.1.1.2. The case where $p^{sel} < 1$ is explored in Sections 2.2.2 and 3.2.3.

This means the parameter estimate, $\hat{\mathbf{a}}$, is $\mathbf{a}^{sel}$, where $\mathbf{a}^{sel}$ denotes the parameter vector on which the selected filter is based. The cost functional becomes

$$J_a^2 \triangleq \frac{\int_{\mathcal{A}} E\{[\mathbf{a}-\mathbf{a}^{sel}]^T \mathbf{W}[\mathbf{a}-\mathbf{a}^{sel}]\} d\mathbf{a}}{\int_{\mathcal{A}} d\mathbf{a}} \tag{75}$$

Note that different filters may be selected at different points in the parameter space.

The following is a procedure by which this cost functional can be numerically approximated and minimized.

1.) Start by describing the system in terms of the parameter vector **a**, specifying the structure of the truth system and the filters to be implemented in the estimator.

2.) Choose K, the number of filters to be implemented in the estimator.

3.) Choose a representative parameter set to begin the minimization. A reasonable choice may be made by equally dividing the admissible parameter region and choosing the midpoint of each region as the representative parameter set.

4.) Choose a numerical integration technique which will be used to evaluate Equation (75) for any given choice of representative parameter set. For parameter minimization it is not necessary to evaluate Equation (48) per se. However, it is still necessary to evaluate Equation (47) to get the information required to find out which filter is selected using Equation (55). The value of $\hat{\mathbf{a}}$ needed in Equation (74) is the value of **a** forming the basis of the filter selected at the evaluation point.

5.) Step 4 generates a numerical approximation to the value J(**a**) for a prespecified choice of parameter vector **a**. The problem then becomes a vector minimization problem. Choose a numerical vector minimization technique and proceed to minimize J(**a**) using the procedure in step 4 to evaluate the functional for each parameter vector.

As is the case with the state estimator, the extended trapezoidal rule [51, Eqn. (4.1.11)] is the numerical integration technique used in the parameter optimization example of Chapter III. The vector minimization is done using a Broyden-Fletcher-Goldfarb-Shanno (BFGS) variable metric method for unconstrained minimization [62] using one-sided difference approximations.

2.2.2 Modifications to the Parameter Estimator Design Algorithm. The parameter estimator optimization algorithm embodies most of the same calculations required in the state estimator optimization algorithm. The only difference is that $\mathrm{tr}[\mathbf{W}\mathbf{\Psi}^{sel-}]$ is replaced by $E\{[\mathbf{a}-\mathbf{a}^{sel}]^{\mathrm{T}}\mathbf{W}[\mathbf{a}-\mathbf{a}^{sel}]\}$ as explained in step 4 above. The algorithm is therefore general in nature and the modifications to parameter optimization algorithm are the same as the modifications to the state estimator algorithm, which are explained in Section 2.1.3, with the exception of placing lower bounds on the probabilities. If $p_{\min}$ denotes the lower bound on the probabilities, the parameter estimate converges to

$$\begin{aligned}
\hat{\mathbf{a}}(t_i^+) &= \sum_{k=1}^{K} \hat{\mathbf{a}}^k(t_i^+)p^k(t_i) && (76)\\
&= \{[1-(K \times p_{\min})]\hat{\mathbf{a}}^{sel}(t_i^+)\} + \sum_{k=1}^{K} \hat{\mathbf{a}}^k(t_i^+)p_{\min} && (77)
\end{aligned}$$

or, more concisely,

$$\mathbf{a}^{MMFA} = \{[1-(K \times p_{\min})]\mathbf{a}^{sel}\} + \sum_{k=1}^{K} \mathbf{a}^k p_{\min} \tag{78}$$

We can modify our calculation of the cost functional by substituting this expression for $\mathbf{a}^{sel}$ in Equation (75) and computing the cost for any given lower bound on the probability.

2.3 Controller Cost Functional

In the next few sections, the equations required to design an optimal MMCA system are developed. The system of interest is a linear regulator as shown in Figure 10 (which is a repeat of Figure 4). In an effort to get good regulation, it is desirable to minimize the cost functional

$$\mathrm{J}_c^{\mathbf{a}} \triangleq \frac{\int_{\mathcal{A}} \mathrm{E}\{[\mathrm{y}]^{\mathsf{T}}\mathbf{W}[\mathrm{y}]\}d\mathbf{a}}{\int_{\mathcal{A}} d\mathbf{a}} \tag{79}$$

Since it is assumed that the admissible parameter set is constant for a given problem, it is necessary only to minimize the numerator of Equation (79). Moreover,

$$\begin{aligned}
\mathrm{E}\{[\mathrm{y}]^{\mathsf{T}}\mathbf{W}[\mathrm{y}]\} &= \mathrm{E}\{\mathrm{tr}[[\mathrm{y}]^{\mathsf{T}}\mathbf{W}[\mathrm{y}]]\}\\
&= \mathrm{E}\{\mathrm{tr}[\mathbf{W}[\mathrm{y}][\mathrm{y}]^{\mathsf{T}}]\}\\
&= \mathrm{tr}[\mathbf{W}\mathrm{E}\{[\mathrm{y}][\mathrm{y}]^{\mathsf{T}}\}]\\
&= \mathrm{tr}[\mathbf{W}\mathrm{E}\{\boldsymbol{C}^*[\mathrm{x}][\mathrm{x}]^{\mathsf{T}}\boldsymbol{C}^{*\mathsf{T}}\}]\\
&= \mathrm{tr}[\mathbf{W}\boldsymbol{C}^*\boldsymbol{\Psi}_c\boldsymbol{C}^{*\mathsf{T}}] && (80)
\end{aligned}$$

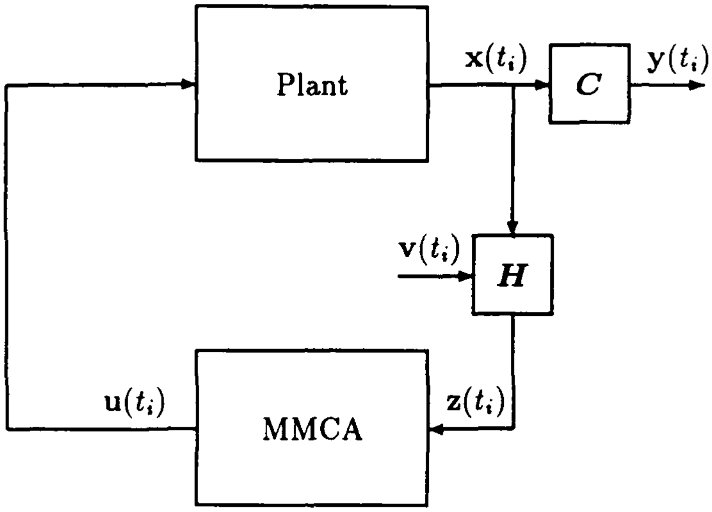


Figure 10. Linear Regulator Using an MMCA

Thus, one needs to develop the equations necessary to compute the regulation error autocorrelation,$\boldsymbol{\Psi}_c = \mathrm{E}\{[\mathrm{x}][\mathrm{x}]^{\mathrm{T}}\}$, as a function of the parameter vector **a**. The following section illustrates this development.

2.3.1 Derivation of the Controller Equations. The MMCA consists of a bank of Kalman filters, each in cascade with a constant gain. Each Kalman filter/controller gain combination is based on a representative parameter vector $\mathbf{a}^k$. We want to find out what the regulation error autocorrelation is when we use a perturbation model based on $\mathbf{a}^k$:

$$\mathbf{x}^k(t_{i+1}) = \boldsymbol{\Phi}^k(t_{i+1}, t_i)\mathbf{x}^k(t_i) + \boldsymbol{B}_d^k(t_i)\mathbf{u}^k(t_i) + \boldsymbol{G}_d^k(t_i)\mathbf{w}_d^k(t_i) \tag{81}$$

$$\mathbf{y}^k(t_i) = \boldsymbol{C}^k(t_i)\mathbf{x}^k(t_i) \tag{82}$$

$$\mathbf{z}^k(t_i) = \boldsymbol{H}^k(t_i)\mathbf{x}^k(t_i) + \mathbf{v}^k(t_i) \tag{83}$$

which varies from the actual system with the truth model which is based on the true parameter vector $\mathbf{a}^*$:

$$\mathbf{x}^*(t_{i+1}) = \boldsymbol{\Phi}^*(t_{i+1}, t_i)\mathbf{x}^*(t_i) + \boldsymbol{B}_d^*(t_i)\mathbf{u}^*(t_i) + \boldsymbol{G}_d^*(t_i)\mathbf{w}_d^*(t_i) \tag{84}$$

$$\mathbf{y}^*(t_i) = \boldsymbol{C}^*(t_i)\mathbf{x}^*(t_i) \tag{85}$$

$$\mathbf{z}^*(t_i) = \boldsymbol{H}^*(t_i)\mathbf{x}^*(t_i) + \mathbf{v}^*(t_i) \tag{86}$$

and where

$\mathbf{x}^k(t_i)$ is the n^k dimensional state vector in the estimator filter models.

$\mathbf{x}^*(t_i)$ is the n^* dimensional error state vector in the truth model; n^* need not be the same as n^k.

$\boldsymbol{\Phi}^k(t_{i+1}, t_i)$ is the $n^k \times n^k$ state transition matrix for the k^{th} filter model.

$\boldsymbol{\Phi}^*(t_{i+1}, t_i)$ is the $n^* \times n^*$ state transition matrix in the truth system model.

$\mathbf{u}^k(t_i)$ is the$p$ dimensional input vector generated by the k^{th} controller.

$\mathbf{u}^*(t_i)$ is the p dimensional input vector which is calculated by

$$\mathbf{u}^*(t_i) \triangleq \sum_{k=1}^{K} \mathbf{u}^k(t_i)p^k(t_i) \tag{87}$$

$\boldsymbol{B}_d^k(t_i)$ is the $n^k \times p$ input matrix for the k^{th} model.

$\boldsymbol{B}_d^*(t_i)$ is the $n^* \times p$ truth system input matrix.

$\mathbf{w}_d^k(t_i)$ is an s^k-vector-valued discrete-time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{[\mathbf{w}_d^k(t_i)][\mathbf{w}_d^k(t_j)]^{\mathsf{T}}\} = \begin{cases} \boldsymbol{Q}_d^k(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases}$$

$\mathbf{w}_d^*(t_i)$ is an s^*-vector-valued discrete-time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{[\mathbf{w}_d^*(t_i)][\mathbf{w}_d^*(t_j)]^{\mathsf{T}}\} = \begin{cases} \boldsymbol{Q}_d^*(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases}$$

$\boldsymbol{G}_d^k(t_i)$ is the $n^k \times s^k$ noise input matrix for the k^{th} filter model.

$G_d^*(t_i)$ is the $n^* \times s^*$ noise input matrix for the truth model.

$\mathbf{y}^k(t_i)$ is the$q$ dimensional output vector to be controlled as modelled in the k^{th} estimator filter model.

$\mathbf{y}^*(t_i)$ is the q dimensional output vector to be controlled in the truth model.

$C^k(t_i)$ is the $q \times n^k$ output matrix for the k^{th} model.

$C^*(t_i)$ is the $q \times n^*$ truth system output matrix.

$\mathbf{z}^k$ is the m dimensional random measurement vector as modelled in the filters.

$\mathbf{z}^*$ is the m dimensional random vector which represents the "real world" measurement from the truth model.

$H^k(t_i)$ is the $m \times n^k$ measurement matrix as modelled in the filters.

$H^*(t_i)$ is the $m \times n^*$ measurement matrix in the truth system model.

$\mathbf{v}^k(t_i)$ is an m-vector-valued discrete-time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{[\mathbf{v}^k(t_i)][\mathbf{v}^k(t_j)]^{\mathsf{T}}\} = \begin{cases} \boldsymbol{R}^k(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases}$$

$\mathbf{w}_d^k$ and $\mathbf{v}^k$ are assumed to be independent.

$\mathbf{v}^*(t_i)$ is an m-vector-valued discrete-time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{[\mathbf{v}^*(t_i)][\mathbf{v}^*(t_j)]^{\mathsf{T}}\} = \begin{cases} \boldsymbol{R}^*(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases}$$

$\mathbf{w}_d^*$ and $\mathbf{v}^*$ are assumed to be independent.

The development uses steady state arguments so the following assumptions are made about time invariance:

$$t_{i+1} - t_i = \text{a constant} \quad \forall i$$

$$\boldsymbol{\Phi}^k(t_{i+1}, t_i) = \boldsymbol{\Phi}^k \quad \forall t$$

$$\boldsymbol{\Phi}^*(t_{i+1}, t_i) = \boldsymbol{\Phi}^* \quad \forall t$$

$$\boldsymbol{B}_d^k(t_i) = \boldsymbol{B}_d^k \quad \forall t$$

$$\boldsymbol{B}_d^*(t_i) = \boldsymbol{B}_d^* \quad \forall t$$

$$\boldsymbol{G}_d^k(t_i) = \boldsymbol{G}_d^k \quad \forall t$$

$$\boldsymbol{G}_d^*(t_i) = \boldsymbol{G}_d^* \quad \forall t$$

$$\boldsymbol{C}^k(t_i) = \boldsymbol{C}^k \quad \forall t$$

$$\boldsymbol{C}^*(t_i) = \boldsymbol{C}^* \quad \forall t$$

$$\boldsymbol{H}^k(t_i) = \boldsymbol{H}^k \quad \forall t$$

$$\boldsymbol{H}^*(t_i) = \boldsymbol{H}^* \quad \forall t$$

The filters are assumed to be constant-gain filters (i.e., $\boldsymbol{K}^k$ does not vary with time). The filter gain may be calculated in any fashion that produces a stable estimator. However, to use this method effectively for minimizing the error autocorrelation, an automated method which can determine $\boldsymbol{K}^k$ for any given parameter is suggested. For the example shown in the sequel, the steady state Kalman filter gain is chosen.

We will similarly assume the controllers are designed in any suitable fashion as constant gain state feedback controllers and implemented with the form

$$\mathrm{u}^k(t_i) = -\boldsymbol{G}_c^k \hat{\mathbf{x}}^k(t_i^+) \tag{88}$$

Again, to use this method effectively, an automated method which can determine $\boldsymbol{G}_c^k$ for any given parameter is suggested. For the example shown in the sequel, an optimal LQG regulator gain [37, Chap. 14] is chosen.

We want to minimize the true output autocorrelation, hence, we want to find the autocorrelation matrix:

$$\mathrm{E}\{[\mathbf{y}_i^*][\mathbf{y}_i^*]^{\mathrm{T}}\} = \boldsymbol{C}^*[\mathrm{E}\{[\mathbf{x}_i^*][\mathbf{x}_i^*]^{\mathrm{T}}\}]\boldsymbol{C}^{*\mathrm{T}} \tag{89}$$

The truth system state is not affected by measurement updates in the MMCA so $\mathbf{x}^*(t_i) = \mathbf{x}^*(t_i^-)$. The filter selection is driven by the prediction error so we will derive the equations for the autocorrelation matrix:

$$\mathrm{E}\{[\mathbf{y}_i^*][\mathbf{y}_i^*]^{\mathrm{T}}\} = \boldsymbol{C}^*[\mathrm{E}\{[\mathbf{x}_i^{*-}][\mathbf{x}_i^{*-}]^{\mathrm{T}}\}]\boldsymbol{C}^{*\mathrm{T}} \tag{90}$$

As was done with the estimator equations, we derive the one-step prediction model from the standard Kalman filter equations [35, page 275] as follows:

$$\hat{\mathbf{x}}^k(t_i^+) = \hat{\mathbf{x}}^k(t_i^-) + \boldsymbol{K}^k[\mathbf{z}^*(t_i) - \boldsymbol{H}^k\hat{\mathbf{x}}^k(t_i^-)] \tag{91}$$

$$\hat{\mathbf{x}}^k(t_{i+1}^-) = \boldsymbol{\Phi}^k\hat{\mathbf{x}}^k(t_i^+) + \boldsymbol{B}_d^k\mathbf{u}^*(t_i) \tag{92}$$

The steady state development requires an important assumption at this point. It is assumed that the estimator section of the MMCA has converged to the proper filter. Hence,

$$p^{sel} = 1 \implies \mathbf{u}^*(t_i) = \mathbf{u}^{sel}(t_i) \tag{93}$$

where *sel* denotes the filter selected by the identification process. All other p^j's are zero. As the functional is integrated across the parameter space, different filters are selected. The following development is for the filter k which has been selected by the identification process, and is not applicable to the other filters. All filters are propagated based on the input vector, $\mathbf{u}^*(t_i)$, not the input calculated by the corresponding elemental controller.

With this in mind, substituting Equation (88) into Equation (92) gives

$$\hat{\mathbf{x}}^k(t_{i+1}^-) = \boldsymbol{\Phi}^k\hat{\mathbf{x}}^k(t_i^+) - \boldsymbol{B}_d^k\boldsymbol{G}_c^k\hat{\mathbf{x}}^k(t_i^+) \tag{94}$$

Substituting Equation (91) into Equation (94) gives

$$\hat{\mathbf{x}}^k(t_{i+1}^-) = (\boldsymbol{\Phi}^k - \boldsymbol{B}_d^k\boldsymbol{G}_c^k)\{\hat{\mathbf{x}}^k(t_i^-) + \boldsymbol{K}^k[\boldsymbol{H}^*\mathbf{x}^*(t_i) + \mathbf{v}^*(t_i) - \boldsymbol{H}^k\hat{\mathbf{x}}^k(t_i^-)]\} \tag{95}$$

Collecting terms yields

$$\begin{aligned}\hat{\mathbf{x}}^k(t_{i+1}^-) &= (\boldsymbol{\Phi}^k - \boldsymbol{B}_d^k\boldsymbol{G}_c^k)(\boldsymbol{I} - \boldsymbol{K}^k\boldsymbol{H}^k)\hat{\mathbf{x}}^k(t_i^-) + (\boldsymbol{\Phi}^k - \boldsymbol{B}_d^k\boldsymbol{G}_c^k)\boldsymbol{K}^k\boldsymbol{H}^*\mathbf{x}^*(t_i) \\ &\quad + (\boldsymbol{\Phi}^k - \boldsymbol{B}_d^k\boldsymbol{G}_c^k)\boldsymbol{K}^k\mathbf{v}^*(t_i) \qquad (96)\end{aligned}$$

The true system is modelled by

$$\mathbf{x}^*(t_{i+1}^-) = \boldsymbol{\Phi}^*\mathbf{x}^*(t_i^-) + \boldsymbol{B}_d^*\mathbf{u}^*(t_i) + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i) \qquad (97)$$

Recalling that $\mathbf{u}^*(t_i) = \mathbf{u}^k(t_i)$, substituting Equation (88) into Equation (97) yields

$$\mathbf{x}^*(t_{i+1}^-) = \boldsymbol{\Phi}^*\mathbf{x}^*(t_i^-) - \boldsymbol{B}_d^*\boldsymbol{G}_c^k\hat{\mathbf{x}}^k(t_i^+) + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i) \qquad (98)$$

Substituting Equation (91) into Equation (98) gives

$$\begin{aligned}\mathbf{x}^*(t_{i+1}^-) &= \boldsymbol{\Phi}^*\mathbf{x}^*(t_i^-) - \boldsymbol{B}_d^*\boldsymbol{G}_c^k\{\hat{\mathbf{x}}^k(t_i^-) + \boldsymbol{K}^k[\boldsymbol{H}^*\mathbf{x}^*(t_i) + \mathbf{v}^*(t_i) - \boldsymbol{H}^k\hat{\mathbf{x}}^k(t_i^-)]\} \\ &\quad + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i) \qquad (99)\end{aligned}$$

Collecting terms yields

$$\begin{aligned}\mathbf{x}^*(t_{i+1}^-) &= (\boldsymbol{\Phi}^* - \boldsymbol{B}_d^*\boldsymbol{G}_c^k\boldsymbol{K}^k\boldsymbol{H}^*)\mathbf{x}^*(t_i^-) - \boldsymbol{B}_d^*\boldsymbol{G}_c^k(\boldsymbol{I} - \boldsymbol{K}^k\boldsymbol{H}^k)\hat{\mathbf{x}}^k(t_i^-) \\ &\quad - \boldsymbol{B}_d^*\boldsymbol{G}_c^k\boldsymbol{K}^k\mathbf{v}^*(t_i) + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i) \qquad (100)\end{aligned}$$

This gives us the one-step prediction model

$$\begin{aligned}\begin{bmatrix}\mathbf{x}_{i+1}^{*-} \\ \hat{\mathbf{x}}_{i+1}^{k-}\end{bmatrix} &= \begin{bmatrix}(\boldsymbol{\Phi}^* - \boldsymbol{B}_d^*\boldsymbol{G}_c^k\boldsymbol{K}^k\boldsymbol{H}^*) & -\boldsymbol{B}_d^*\boldsymbol{G}_c^k(\boldsymbol{I} - \boldsymbol{K}^k\boldsymbol{H}^k) \\ (\boldsymbol{\Phi}^k - \boldsymbol{B}_d^k\boldsymbol{G}_c^k)\boldsymbol{K}^k\boldsymbol{H}^* & (\boldsymbol{\Phi}^k - \boldsymbol{B}_d^k\boldsymbol{G}_c^k)(\boldsymbol{I} - \boldsymbol{K}^k\boldsymbol{H}^k)\end{bmatrix}\begin{bmatrix}\mathbf{x}_i^{*-} \\ \hat{\mathbf{x}}_i^{k-}\end{bmatrix} \\ &\quad + \begin{bmatrix}-\boldsymbol{B}_d^*\boldsymbol{G}_c^k\boldsymbol{K}^k \\ (\boldsymbol{\Phi}^k - \boldsymbol{B}_d^k\boldsymbol{G}_c^k)\boldsymbol{K}^k\end{bmatrix}\mathbf{v}_i^* + \begin{bmatrix}\boldsymbol{G}_d^* \\ \mathbf{0}\end{bmatrix}\mathbf{w}_{di}^* \qquad (101)\end{aligned}$$

The prediction model of the tracking error equation is now of the form:

$$\begin{bmatrix}\mathbf{x}_{i+1}^{*-} \\ \hat{\mathbf{x}}_{i+1}^{k-}\end{bmatrix} = \mathcal{T}\begin{bmatrix}\mathbf{x}_i^{*-} \\ \hat{\mathbf{x}}_i^{k-}\end{bmatrix} + \mathcal{G}_o\begin{bmatrix}\mathbf{w}_{di}^* \\ \mathbf{v}_i^*\end{bmatrix} \qquad (102)$$

where

$$\mathcal{T} = \begin{bmatrix}(\boldsymbol{\Phi}^* - \boldsymbol{B}_d^*\boldsymbol{G}_c^k\boldsymbol{K}^k\boldsymbol{H}^*) & -\boldsymbol{B}_d^*\boldsymbol{G}_c^k(\boldsymbol{I} - \boldsymbol{K}^k\boldsymbol{H}^k) \\ (\boldsymbol{\Phi}^k - \boldsymbol{B}_d^k\boldsymbol{G}_c^k)\boldsymbol{K}^k\boldsymbol{H}^* & (\boldsymbol{\Phi}^k - \boldsymbol{B}_d^k\boldsymbol{G}_c^k)(\boldsymbol{I} - \boldsymbol{K}^k\boldsymbol{H}^k)\end{bmatrix}$$

$$\mathcal{G}_o = \begin{bmatrix} \boldsymbol{G}_d^* & -\boldsymbol{B}_d^*\boldsymbol{G}_c^k\boldsymbol{K}^k \\ 0 & (\boldsymbol{\Phi}^k - \boldsymbol{B}_d^k\boldsymbol{G}_c^k)\boldsymbol{K}^k \end{bmatrix}$$

and $\begin{bmatrix} \mathbf{w}_{di}^* \\ \mathbf{v}_i^* \end{bmatrix}$ is an $s^* + m$-vector valued discrete time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{\begin{bmatrix} \mathbf{w}_{di}^* \\ \mathbf{v}_i^* \end{bmatrix} [\mathbf{w}_{dj}^{*\mathsf{T}}\mathbf{v}_j^{*\mathsf{T}}]\} = \begin{cases} \boldsymbol{Q}_o(t_i) & t_i = t_j \\ 0 & t_i \neq t_j \end{cases}$$

with

$$\boldsymbol{Q}_o(t_i) = \begin{bmatrix} \boldsymbol{Q}_d^*(t_i) & 0 \\ 0 & \boldsymbol{R}^*(t_i) \end{bmatrix}$$

Since $\mathbf{w}_d^*$ and $\mathbf{v}^*$ are assumed to be independent random vectors, they are uncorrelated.

This structure allows us to calculate the autocorrelation of $\begin{bmatrix} \mathbf{x}_{i+1}^{*-} \\ \hat{\mathbf{x}}_{i+1}^{sel-} \end{bmatrix}$ using the matrix equation

$$\mathrm{E}\left\{\begin{bmatrix} \mathbf{x}_{i+1}^{*-} \\ \hat{\mathbf{x}}_{i+1}^{sel-} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{i+1}^{*-\mathsf{T}} & \hat{\mathbf{x}}_{i+1}^{sel-\mathsf{T}} \end{bmatrix}\right\} = \boldsymbol{\Xi}_{c\,i+1} = \boldsymbol{\mathcal{T}}\boldsymbol{\Xi}_{ci}\boldsymbol{\mathcal{T}}^{\mathsf{T}} + \mathcal{G}_o\boldsymbol{Q}_o\mathcal{G}_o^{\mathsf{T}} \qquad (103)$$

where *sel* denotes the filter selected at any particular point in the admissible parameter set so that $p^{sel} = 1$. The case where $p^{sel} < 1$ is discussed in Sections 2.3.3.3 and 3.3.5.4.

The upper left partition, $\mathrm{E}\{[\mathbf{x}_i^{*-}][\mathbf{x}_i^{*-}]^{\mathsf{T}}\}$, of this matrix is $\boldsymbol{\Psi}_c$. Letting $i \Rightarrow \infty$, we can calculate the steady state value of $\mathrm{E}\{[\mathbf{y}^{*-}]^{\mathsf{T}}\mathbf{W}[\mathbf{y}^{*-}]\}$ using Equation (80).

2.3.2 Controller Design Algorithm. The following is a procedure by which the cost functional of Equation (79) can be numerically approximated and minimized to provide the representative parameters for an optimized multiple model adaptive regulator. The algorithm may be adapted for other control structures such as trackers [37, pages 114-122] or perturbation regulators [37, page 125].

1.) Start by describing the system in terms of the parameter vector **a**, specifying the structure of the truth system and the filters to be implemented in the controller. This includes specifying the method for calculating $\boldsymbol{K}^k$ and $\boldsymbol{G}_c^k$ given the parameter vector **a**.

2.) Choose K, the number of regulators to be implemented.

3.) Choose a representative parameter set to begin the minimization. A reasonable choice might be made by dividing the admissible parameter region into equal intervals and choosing the midpoint of each interval. The regulator problem is more likely to produce unstable systems than the estimator problem. Even though the LQ regulator has guaranteed stability margins [37, page 11], and the Kalman filter has similar robustness [37, page 99], the filter/controller combination has no guaranteed stability margins [9]. If the starting point results in an unstable regulator for a particular parameter value, the designer has the choice of changing the rule for the controller gain calculations to make the regulators more robust, or moving the closest filter towards the unstable point. It might also be desirable to try the minimization with more filters to gain insight for systems with fewer filters. One other method that worked particularly well for the example problem of Section 3.3.2, was using a penalty function that places an additional cost on each unstable point evaluated while calculating the cost.

4.) Choose a numerical integration technique which will be used to evaluate Equation (79). This integration provides the value of the functional for a specific choice of representative parameter set. Equation (80) defines the curve of regulation error autocorrelation as a function of the parameter vector. The information required to set up Equations (101) and (103) is chosen based on the proper filter selection per Equation (55). It should be noted that the filter selection is based entirely on the estimator in the loop, and is calculated with the equations developed in Section 2.1.1.

5.) Step 4 generates a numerical approximation to the value J(**a**) for a prespecified choice of parameter vector **a**. The problem then becomes a vector minimization problem. Choose a numerical vector minimization technique and proceed to minimize J(**a**) using the procedure in step 4 to evaluate the functional for each parameter vector.

As is the case with the state estimator example, the extended trapezoidal rule [51, Eqn. (4.1.11)] is the numerical integration technique used in the regulator example of Chapter III. The vector minimization is done using a Broyden-Fletcher-Goldfarb-Shanno (BFGS) variable metric method for unconstrained minimization [62].

2.3.3 Modifications to the Controller Design Algorithm. The development of the controller optimization algorithm is analogous to the development of the state estimator optimization algorithm. So, as explained in Section 2.1.3, the algorithm is general enough to account for MIMO systems and parameter vectors. Incorporating the ME/I mechanization is accomplished in the same manner as it in the state estimator optimization algorithm.

2.3.3.1 Controller Weighting Matrices. The controller optimization algorithm also incorporates a weighting matrix, **W**, to allow the designer to handle tradeoffs between states. In addition, the control problem provides added flexibility to vary the respective weight of the states. The controller gains are calculated to minimize a continuous time quadratic cost functional of the form

$$\mathcal{J} = \int_0^\infty [\mathbf{x}^\mathrm{T}(t)\mathbf{S}\mathbf{x}(t) + \mathbf{u}^\mathrm{T}(t)\mathbf{U}\mathbf{u}(t)]\mathrm{d}t \tag{104}$$

Note that different choices of **S** and **U** produce different gain matrices, G_c^k. The effect of varying these weighting matrices is explored in Section 3.3.5.1, where these matrices are chosen to be consistent with the choice of **W**. Maintaining this consistency is not necessarily required for the algorithm to work properly.

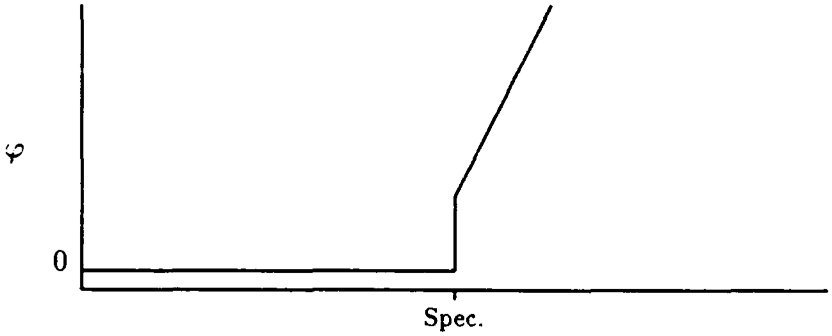
2.3.3.2 An Alternative Maximum-Type Cost Functional. As shown in Section 2.1.3.2, a maximum-type cost may be added to the basic quadratic cost functional, allowing the designer to trade off between average mean square regulation error and maximum mean square regulation error. Many times, however, the maximum allowable mean square regulation error is set to a specific value, i.e.,

$$\max_{\mathbf{a}\in\mathcal{A}} \Psi_c = \Psi_{c\ \mathbf{max}} \leq \text{Spec}$$

The basic controller optimization algorithm can be modified to attack this type of specification by adding a penalty function, φ , to the quadratic cost so that

$$\mathrm{J} \triangleq \mathrm{J}_c^2 + \varphi \tag{105}$$

The penalty function, φ, is a function of the maximum regulation error autocorrelation as shown in Figure 11.



Maximum Regulation Error Autocorrelation, $\Psi_{c\ \mathbf{max}}$

Figure 11. Penalty Function for Maximum Regulation Error Autocorrelation

The penalty function adds a quantum cost if $\Psi_{c\ \mathbf{max}}$ is above the specification and increases linearly as $\Psi_{c\ \mathbf{max}}$ increases. The quantum cost should be large enough so that it overwhelms the quadratic component of the cost functional when $\Psi_{c\ \mathbf{max}}$ exceeds the specifications, thus, prohibiting tradeoff between the penalty function φ and the quadratic cost J_c^2. The linear region gives the optimization algorithm information concerning which direction to go in order to decrease the cost. This method is used in Section 3.3.5.3.

2.3.3.3 Avoiding Controller Lockup. The development in Section 2.3.1 was based upon the assumption that the MMCA had converged to the selected controller with probability 1. It has been shown in Section 2.1.3.3 that it is reasonable to place lower bounds on the probabilities to keep the estimator section of the MMCA from experiencing lockup. The controller cost functional calculations can be modified to take this into account. To do so, we must derive the one-step prediction model from the standard Kalman filter equations [35, page 275] as follows:

$$\hat{\mathbf{x}}^k(t_i^+) = \hat{\mathbf{x}}^k(t_i^-) + \boldsymbol{K}^k[\mathbf{z}^*(t_i) - \boldsymbol{H}^k\hat{\mathbf{x}}^k(t_i^-)] \tag{106}$$

$$\hat{\mathbf{x}}^k(t_{i+1}^-) = \boldsymbol{\Phi}^k\hat{\mathbf{x}}^k(t_i^+) + \boldsymbol{B}_d^k\mathbf{u}^*(t_i) \tag{107}$$

At this point in the development of Section 2.3.1, it was assumed that the estimator section of the MMCA has converged to the proper filter. Instead, we assume the input is the probabilistically weighted average:

$$\mathbf{u}^*(t_i) = \sum_{j=1}^{K} p^j\mathbf{u}^j(t_i) \tag{108}$$

and each input is of the form

$$\mathbf{u}^j(t_i) = -\boldsymbol{G}_c^j\hat{\mathbf{x}}^j(t_i^+) \tag{109}$$

With this in mind, substituting Equations (108) and (109) into Equation (107) gives

$$\begin{aligned}\hat{\mathbf{x}}^k(t_{i+1}^-) &= \boldsymbol{\Phi}^k\hat{\mathbf{x}}^k(t_i^+) + \boldsymbol{B}_d^k\sum_{j=1}^{K} p^j\mathbf{u}^j(t_i) &(110)\\ &= \boldsymbol{\Phi}^k\hat{\mathbf{x}}^k(t_i^+) - \boldsymbol{B}_d^k\sum_{j=1}^{K} p^j\boldsymbol{G}_c^j\hat{\mathbf{x}}^j(t_i^+) &(111)\end{aligned}$$

Substituting Equation (106) into Equation (111) gives

$$\begin{aligned}\hat{\mathbf{x}}^k(t_{i+1}^-) &= \boldsymbol{\Phi}^k\{\hat{\mathbf{x}}^k(t_i^-) + \boldsymbol{K}^k[\mathbf{z}^*(t_i) - \boldsymbol{H}^k\hat{\mathbf{x}}^k(t_i^-)]\}\\ &\quad - \boldsymbol{B}_d^k\sum_{j=1}^{K} p^j\boldsymbol{G}_c^j\{\hat{\mathbf{x}}^j(t_i^-) + \boldsymbol{K}^j[\mathbf{z}^*(t_i) - \boldsymbol{H}^j\hat{\mathbf{x}}^j(t_i^-)]\} &(112)\end{aligned}$$

$$\begin{aligned}\hat{\mathbf{x}}^k(t_{i+1}^-) &= \boldsymbol{\Phi}^k\{\hat{\mathbf{x}}^k(t_i^-) + \boldsymbol{K}^k[\boldsymbol{H}^*\mathbf{x}^*(t_i) + \mathbf{v}^*(t_i) - \boldsymbol{H}^k\hat{\mathbf{x}}^k(t_i^-)]\}\\ &\quad -\boldsymbol{B}_d^k\sum_{j=1}^{K} p^j\boldsymbol{G}_c^j\hat{\mathbf{x}}^j(t_i^-)\\ &\quad -\boldsymbol{B}_d^k\sum_{j=1}^{K} p^j\boldsymbol{G}_c^j\boldsymbol{K}^j[\boldsymbol{H}^*\mathbf{x}^*(t_i) + \mathbf{v}^*(t_i) - \boldsymbol{H}^j\hat{\mathbf{x}}^j(t_i^-)] &(113)\end{aligned}$$

Rearrange and collect terms to get

$$
\begin{aligned}
\hat{\mathbf{x}}^k(t_{i+1}^-) &= \boldsymbol{\Phi}^k(\boldsymbol{I} - \boldsymbol{K}^k\boldsymbol{H}^k)\hat{\mathbf{x}}^k(t_i^-) + \boldsymbol{\Phi}^k\boldsymbol{K}^k\boldsymbol{H}^*\mathbf{x}^*(t_i) + \boldsymbol{\Phi}^k\boldsymbol{K}^k\mathbf{v}^*(t_i) \\
&\quad - \boldsymbol{B}_d^k \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j(\boldsymbol{I} - \boldsymbol{K}^j\boldsymbol{H}^j)\hat{\mathbf{x}}^j(t_i^-) \\
&\quad - \boldsymbol{B}_d^k \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j\boldsymbol{K}^j\boldsymbol{H}^*\mathbf{x}^*(t_i) \\
&\quad - \boldsymbol{B}_d^k \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j\boldsymbol{K}^j\mathbf{v}^*(t_i)
\end{aligned}
\tag{114}
$$

$$
\begin{aligned}
\hat{\mathbf{x}}^k(t_{i+1}^-) &= \boldsymbol{\Phi}^k(\boldsymbol{I} - \boldsymbol{K}^k\boldsymbol{H}^k)\hat{\mathbf{x}}^k(t_i^-) \\
&\quad - \boldsymbol{B}_d^k \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j(\boldsymbol{I} - \boldsymbol{K}^j\boldsymbol{H}^j)\hat{\mathbf{x}}^j(t_i^-) \\
&\quad + \left(\boldsymbol{\Phi}^k\boldsymbol{K}^k\boldsymbol{H}^* - \boldsymbol{B}_d^k \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j\boldsymbol{K}^j\boldsymbol{H}^*\right)\mathbf{x}^*(t_i) \\
&\quad + \left(\boldsymbol{\Phi}^k\boldsymbol{K}^k - \boldsymbol{B}_d^k \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j\boldsymbol{K}^j\right)\mathbf{v}^*(t_i)
\end{aligned}
\tag{115}
$$

The true system is modelled by

$$
\mathbf{x}^*(t_{i+1}^-) = \boldsymbol{\Phi}^*\mathbf{x}^*(t_i^-) + \boldsymbol{B}_d^*\mathbf{u}^*(t_i) + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i)
\tag{116}
$$

Substituting Equations (108) and (109) into Equation (115) yields

$$
\begin{aligned}
\mathbf{x}^*(t_{i+1}^-) &= \boldsymbol{\Phi}^*\mathbf{x}^*(t_i^-) + \boldsymbol{B}_d^* \sum_{j=1}^{K} p^j\mathbf{u}^j(t_i) + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i) \\
&= \boldsymbol{\Phi}^*\mathbf{x}^*(t_i^-) - \boldsymbol{B}_d^* \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j\hat{\mathbf{x}}^j(t_i^+) + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i)
\end{aligned}
\tag{117}
$$

Substitute Equation (106) into Equation (117) to get

$$
\begin{aligned}
\mathbf{x}^*(t_{i+1}^-) &= \boldsymbol{\Phi}^*\mathbf{x}^*(t_i^-) \\
&\quad - \boldsymbol{B}_d^* \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j\{\hat{\mathbf{x}}^j(t_i^-) + \boldsymbol{K}^j[\mathbf{z}^*(t_i) - \boldsymbol{H}^j\hat{\mathbf{x}}^j(t_i^-)]\} \\
&\quad + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i)
\end{aligned}
\tag{118}
$$

$$\begin{aligned}
\mathbf{x}^*(t_{i+1}^-) &= \boldsymbol{\Phi}^*\mathbf{x}^*(t_i^-) \\
&\quad - \boldsymbol{B}_d^* \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j \{\hat{\mathbf{x}}^j(t_i^-) + \boldsymbol{K}^j[\boldsymbol{H}^*\mathbf{x}^*(t_i) + \mathbf{v}^*(t_i) - \boldsymbol{H}^j\hat{\mathbf{x}}^j(t_i^-)]\} \\
&\quad + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i)
\end{aligned} \tag{119}$$

$$\begin{aligned}
\mathbf{x}^*(t_{i+1}^-) &= (\boldsymbol{\Phi}^* - \boldsymbol{B}_d^* \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j \boldsymbol{K}^j \boldsymbol{H}^*)\mathbf{x}^*(t_i^-) \\
&\quad - \boldsymbol{B}_d^* \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j (\boldsymbol{I} - \boldsymbol{K}^j\boldsymbol{H}^j)\hat{\mathbf{x}}^j(t_i^-) \\
&\quad - \boldsymbol{B}_d^* \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j \boldsymbol{K}^j \mathbf{v}^*(t_i) + \boldsymbol{G}_d^*\mathbf{w}_d^*(t_i)
\end{aligned} \tag{120}$$

This gives us the one-step prediction model of the form:

$$\begin{bmatrix} \mathbf{x}_{i+1}^{*\,-} \\ \hat{\mathbf{x}}_{i+1}^{1\,-} \\ \vdots \\ \hat{\mathbf{x}}_{i+1}^{K\,-} \end{bmatrix} = \mathcal{T}_a \begin{bmatrix} \mathbf{x}_i^{*\,-} \\ \hat{\mathbf{x}}_i^{1\,-} \\ \vdots \\ \hat{\mathbf{x}}_i^{K\,-} \end{bmatrix} + \mathcal{G}_a \begin{bmatrix} \mathbf{w}_{di}^* \\ \mathbf{v}_i^* \end{bmatrix} \tag{121}$$

where

$$\mathcal{T}_a = \left[\begin{array}{cc}
(\boldsymbol{\Phi}^* - \boldsymbol{B}_d^* \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j \boldsymbol{K}^j \boldsymbol{H}^*) & -\boldsymbol{B}_d^* p^1 \boldsymbol{G}_c^1 (\boldsymbol{I} - \boldsymbol{K}^1 \boldsymbol{H}^1) \\
\left(\boldsymbol{\Phi}^1 \boldsymbol{K}^1 \boldsymbol{H}^* - \boldsymbol{B}_d^1 \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j \boldsymbol{K}^j \boldsymbol{H}^*\right) & \boldsymbol{\Phi}^1(\boldsymbol{I} - \boldsymbol{K}^1\boldsymbol{H}^1) - \boldsymbol{B}_d^1 p^1 \boldsymbol{G}_c^1 (\boldsymbol{I} - \boldsymbol{K}^1 \boldsymbol{H}^1) \\
\vdots & \vdots \\
\left(\boldsymbol{\Phi}^K \boldsymbol{K}^K \boldsymbol{H}^* - \boldsymbol{B}_d^K \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j \boldsymbol{K}^j \boldsymbol{H}^*\right) & -\boldsymbol{B}_d^K p^1 \boldsymbol{G}_c^1 (\boldsymbol{I} - \boldsymbol{K}^1 \boldsymbol{H}^1)
\end{array} \right.$$

$$\left. \begin{array}{cc}
\cdots & -\boldsymbol{B}_d^* p^K \boldsymbol{G}_c^K (\boldsymbol{I} - \boldsymbol{K}^K \boldsymbol{H}^K) \\
\cdots & -\boldsymbol{B}_d^1 p^K \boldsymbol{G}_c^K (\boldsymbol{I} - \boldsymbol{K}^K \boldsymbol{H}^K) \\
\ddots & \vdots \\
\cdots & \boldsymbol{\Phi}^K(\boldsymbol{I} - \boldsymbol{K}^K\boldsymbol{H}^K) - \boldsymbol{B}_d^K p^K \boldsymbol{G}_c^K (\boldsymbol{I} - \boldsymbol{K}^K \boldsymbol{H}^K)
\end{array} \right]$$

and

$$\mathcal{G}_a = \begin{bmatrix}
\boldsymbol{G}_d^* & -\boldsymbol{B}_d^* \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j \boldsymbol{K}^j \\
0 & \left(\boldsymbol{\Phi}^1 \boldsymbol{K}^1 - \boldsymbol{B}_d^1 \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j \boldsymbol{K}^j\right) \\
\vdots & \vdots \\
0 & \left(\boldsymbol{\Phi}^K \boldsymbol{K}^K - \boldsymbol{B}_d^K \sum_{j=1}^{K} p^j \boldsymbol{G}_c^j \boldsymbol{K}^j\right)
\end{bmatrix}$$

and $\begin{bmatrix} \mathbf{w}_{di}^{*} \\ \mathbf{v}_{i}^{*} \end{bmatrix}$ is an $s^{*} + m$-vector valued discrete time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{\begin{bmatrix} \mathbf{w}_{di}^{*} \\ \mathbf{v}_{i}^{*} \end{bmatrix} [\mathbf{w}_{dj}^{*\mathsf{T}} \mathbf{v}_{j}^{*\mathsf{T}}]\} = \begin{cases} \boldsymbol{Q}_{\mathrm{o}}(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases}$$

with

$$\boldsymbol{Q}_{\mathrm{o}}(t_i) = \begin{bmatrix} \boldsymbol{Q}_{d}^{*}(t_i) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R}^{*}(t_i) \end{bmatrix}$$

Since $\mathbf{w}_{d}^{*}$ and $\mathbf{v}^{*}$ are assumed to be independent random vectors, they are uncorrelated.

This stucture allows us to calculate the autocorrelation of $\begin{bmatrix} \mathbf{x}_{i}^{*-} \\ \hat{\mathbf{x}}_{i}^{1-} \\ \vdots \\ \hat{\mathbf{x}}_{i}^{K-} \end{bmatrix}$ using the matrix equation

$$\mathrm{E}\left\{ \begin{bmatrix} \mathbf{x}_{i+1}^{*-} \\ \hat{\mathbf{x}}_{i+1}^{1-} \\ \vdots \\ \hat{\mathbf{x}}_{i+1}^{K-} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{i+1}^{*-\mathsf{T}} & \hat{\mathbf{x}}_{i+1}^{1-\mathsf{T}} & \cdots & \hat{\mathbf{x}}_{i+1}^{K-\mathsf{T}} \end{bmatrix} \right\} = \boldsymbol{\Xi}_{a\,i+1} = \mathcal{T}_a \boldsymbol{\Xi}_{a\,i} \mathcal{T}_a^{\mathsf{T}} + \mathcal{G}_a \boldsymbol{Q}_{\mathrm{o}} \mathcal{G}_a^{\mathsf{T}} \tag{122}$$

The upper left partition, $\mathrm{E}\{[\mathbf{x}_i^{*-}][\mathbf{x}_i^{*-}]^{\mathsf{T}}\}$, of this matrix is denoted $\boldsymbol{\Psi}_{\mathbf{c}}^{-}$. $\mathrm{E}\{[\mathbf{y}_i^{*-}]^{\mathsf{T}}\mathbf{W}[\mathbf{y}_i^{*-}]\}$ can then be calculated using Equation (80).

Using Equation (122) instead of Equation (103) allows us to calculate the cost of a sys.em for any given lower bound on the probability. Note that if the lower bound is zero, Equation (103) reduces to Equation (122).

2.4 Summary

The development of the basic optimization algorithms for MMFA and MMCA design has been presented in this chapter. These algorithms minimize the prediction error autocorrelation at t_i^- with the anticipation that the error autocorrelation at t_i^+ is also minimized. The algorithms as presented are general in nature and applicable for any finite system. The only limitation to practical design is the amount of computing power available to perform the calculations required for the algorithm. A weighting matrix in the cost functional allows the designer to vary the importance of individual states or parameters.

Several possible modifications to the basic MMFA and MMCA structure have been proposed which affect the performance of the final systems. These include changing the hypothesis conditional probability calculation, using alternative cost functionals, and placing lower bounds on the pseudo-probabilities. The development of the associated modifications to the optimization algorithms have also been presented.

An example is presented in Chapter III to illustrate the benefits of using these algoiithms; in part, to analyze multiple model adaptive estimation and control systems; and as a whole, to design MMFAs and MMCAs to provide minimum mean square errors. The modifications discussed above are also evaluated in Chapter III.

III. Evaluation

The system design methods presented in the previous chapter seem to be sound. However, with any design method it is necessary to have some quantitative information pertaining to the value of the method for "real world" design. It is impossible to provide a numerical value representing the increase in performance over other design methods without first knowing the specifics about the problem at hand. A very simple example is presented in this chapter to help the reader evaluate the merits of this design method. The example problem is described in Section 3.1 along with the MMFA evaluation. The same problem is used to evaluate the parameter estimation performance in Section 3.2, and the MMCA regulator performance in Section 3.3.

3.1 State Estimator Evaluation

An example is shown here to demonstrate how useful insights can be gained from approaching the design of an MMFA through minimizing the prediction error. A variety of previously developed methods are used to design estimators, and their associated cost and parameter discretization are compared to the cost and discretization of an estimator which minimizes the cost functional, Equation (9). The exact procedure used to calculate the functional and to find a minimizing set of representative filters is explained in Section 3.1.2.

3.1.1 System Description. The true system is an ideal mechanical translational system as shown in Figure 12. It is a continuous time system which is modelled by the second order linear equation:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\,\mathbf{x}(t) + \mathbf{B}(t)\,\mathbf{u}(t) \tag{123}$$

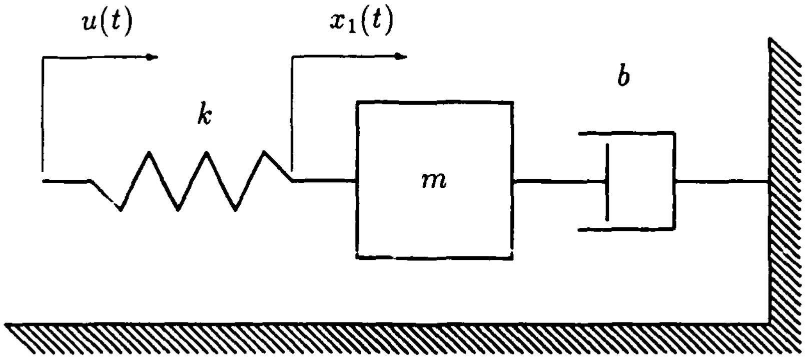


Figure 12. Ideal Mechanical Translational System

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k}{m} \end{bmatrix} u(t) \tag{124}$$

where x_1 is the position in meters and x_2 is the velocity in meters per second. This problem can be restated in a more general form by letting ω_n denote the undamped natural frequency, $\sqrt{\frac{k}{m}}$, and letting ζ denote the damping ratio, $\frac{b}{2\sqrt{km}}$.

To frame the system in a stochastic setting, a dynamic driving noise term is added to the model to produce

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{125}$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} w(t) \tag{126}$$

with

$$\mathrm{E}\{w(t)w(t+\tau)\} = Q\delta(\tau)$$

and

$$Q = 0.01$$

The damping ratio is known to be$\zeta = 0.01$ and the natural frequency,ω_n , is allowed to vary from 2π ($\sim$ 6.28) to 20π ($\sim$ 62.83) radians per second. These values of Q, ζ, and ω_n were chosen to accentuate the differences between models by increasing the sensitivity to change in natural frequency. Larger damping ratios such as $\zeta = 0.707$ result in a system where there is less distinguishable difference between models, so a multiple model technique may not be warranted at all. Q was chosen to provide enough system energy for convergence of the adaptation process, yet not so much as to swamp out the system with noise effects. The values chosen may be associated with a large space structure and are not unrealistic [23, page 44]. Any system with a high sensitivity to parameter variation may be a candidate for an MMFA.

The system is implemented with an equivalent discrete time system model of the form [35, page 171]:

$$\mathbf{x}(t_{i+1}) = \boldsymbol{\Phi}(t_{i+1}, t_i)\mathbf{x}(t_i) + \boldsymbol{B}_d(t_i)\mathbf{u}(t_i) + \mathbf{w}_d(t_i) \tag{127}$$

where

$$\begin{aligned} t_{i+1} - t_i &= 0.01 \text{ sec } \forall\ i \\ \boldsymbol{\Phi}(t_{i+1}, t_i) &= \text{the state transition matrix associated with } \mathbf{F}(t) \\ \boldsymbol{B}_d(t_i) &= \int_{t_i}^{t_{i+1}} \boldsymbol{\Phi}(t_{i+1}, \tau)\mathbf{B}(\tau)\mathrm{d}\tau \end{aligned}$$

and $\mathbf{w}_d(t_i)$ is discrete-time white Gaussian noise process with zero mean and covariance kernel

$$\mathrm{E}\{[\mathbf{w}_d(t_i)][\mathbf{w}_d(t_j)]^{\mathsf{T}}\} = \begin{cases} \boldsymbol{Q}_d(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases}$$

where

$$\boldsymbol{Q}_d(t_i) = \int_{t_i}^{t_{i+1}} \boldsymbol{\Phi}(t_{i+1}, \tau)\mathbf{G}(\tau)Q(\tau)\mathbf{G}^{\mathsf{T}}(\tau)\boldsymbol{\Phi}^{\mathsf{T}}(t_{i+1}, \tau)\mathrm{d}\tau$$

Noisy measurements are available at each sample instant described by the equation

$$z(t_i) = \boldsymbol{H}(t_i)\mathbf{x}(t_i) + v(t_i) \tag{128}$$

$$z(t_i) = [1 \quad 0]\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + v(t_i) \tag{129}$$

where $v(t_i)$ is a zero-mean white Gaussian noise process with covariance kernel

$$\mathrm{E}\{v(t_i)v(t_j)\} = \begin{cases} R(t_i) & i = j \\ 0 & i \neq j \end{cases}$$

and

$$R(t_i) = 0.01 \; \forall \, i$$

This is the same problem Hentz studied, except it is simplified for illustration by not allowing ζ to vary [19, pages 16-1⁰], [39].

3.1.2 Filter Selection Process. This section describes in detail the steps involved in designing an MMFA for the example problem. The main point of this section is to show how useful insights can be gained by attacking an estimator design problem through minimizing the average prediction error autocorrelation.

The first step in any design is to find out what the optimal or best result looks like. The optimal estimation error autocorrelation corresponds to an MMFA consisting of an uncountably infinite number of filters, each designed with ω_n at a different point in the interval $[2\pi, 20\pi]$ and all other values equal to the truth system values [36, pages 129-130]. One can approximate this solution by calculating the cost functional for a fine discretization of the parameter space assuming that there is a filter designed at each evaluated point. This involves solving Equation (47) at a sufficient number of points in the interval to produce a smooth curve. Figure 13 is a plot of the trace of the minimum achievable steady state error autocorrelation matrix,$\mathbf{\Psi}_o$, for the example problem as a function of the true natural frequency, using a discretization of 201 equally spaced points.

This plot of the optimal estimation error provides valuable information about the performance one can expect from an MMFA. The prediction error covariance of an implementable MMFA will always be above this curve. The shape of the curve also provides information for a first guess at the optimized design. As shown in the next section, the filter parameters should be placed closer together at higher values

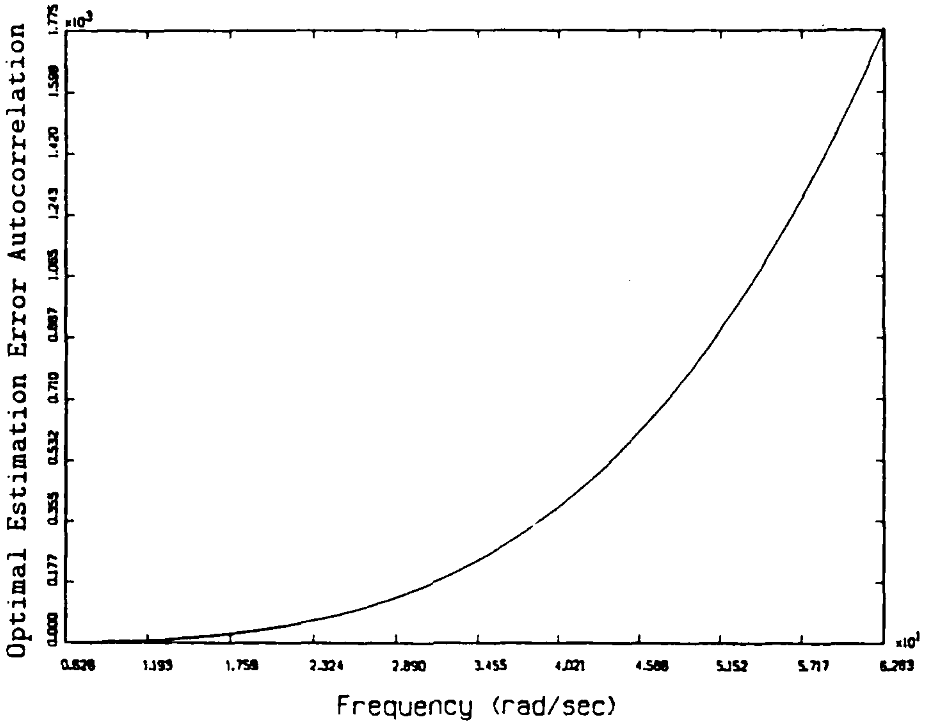
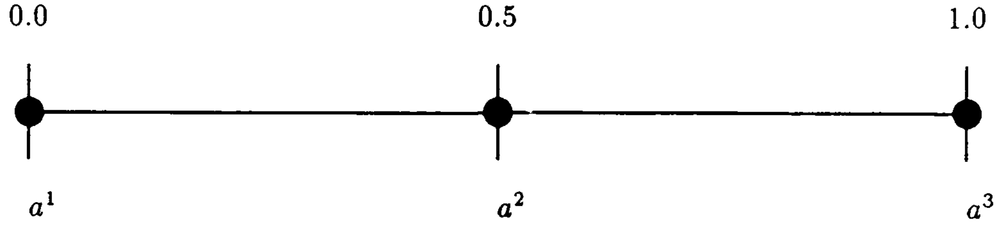


Figure 13. Optimal Estimation Error Autocorrelation for Example Problem

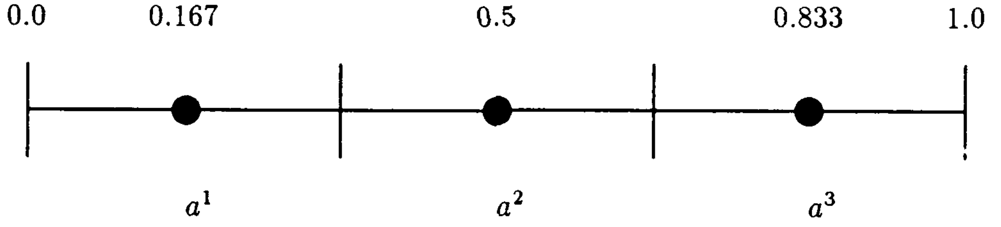
of $\mathrm{tr}\Psi_o$, because the system is more sensitive to change in the parameter at higher values of $\mathrm{tr}\Psi_o$. The closer the starting point is to the optimal design, the fewer evaluations will be necessary to get within the desired accuracy.

The value of the cost functional for the example problem, with $\mathbf{W} = \mathbf{I}_{n \times n}$ (giving position and velocity equal weight) and assuming an infinite bank of filters is used, can be calculated by integrating the curve in Figure 13. This value is denoted J_o and is calculated to be $0.04393\mathrm{m}^2 + 455.5\mathrm{m}^2/\mathrm{s}^2$. However, it is obvious that no one can build an estimator with an infinite number of filters. Several methods have been used previously to select K representative parameter values for the estimator. The methods applicable to this problem are listed here.

Method 1. An uninformed first choice might be to divide the parameter space into K-1 intervals and design a filter at each interval boundary [10, 23, 24]. (e.g., chosing three design parameters from the interval [0,1] using this method would produce the parameters (0.0,0.5,1.0).) Hentz did this with ζ in his thesis [19, pages 19-20].



Method 2. A method designed to cover the parameter space better than method 1 is to divide the parameter space linearly into K intervals and design a filter at the midpoint of each interval. (e.g., chosing three design parameters from the interval [0,1] using this method would produce the parameters (0.1667,0.5,0.8333).)



Method 3. Hentz noticed that frequency response characteristics vary logarithmically in this system so he divided the ω_n space logarithmically into K-1 intervals and designed a filter at each boundary [19, pages 19-20]. Hentz divided the ω_n space into nine intervals and designed ten filters using the boundary points. Table 1 lists the values he used for his bank of filters.

Method 4. The latest method to be used at AFIT is to choose a nominal point to design the first filter and simulate responses as the true parameter value moves away from the nominal, choosing an arbitrary reduction in estimation accuracy to signal the point at which a new filter should be designed (see [29, Section 4.4.4] and [30, 34]). For example, a 45% increase in the value of $E\{[\tilde{\mathbf{x}}(t_i^+)]^T[\tilde{\mathbf{x}}(t_i^+)]\}$ (refer to Eqn. (29)) is used in the example so that three filters cover the region.

Filter	ω_n (rad/sec)
1	6.28
2	8.12
3	10.48
4	13.54
5	17.48
6	22.58
7	29.16
8	37.67
9	48.65
10	62.83

Table 1. Logarithmic Division of the Frequency Parameter

	ω_n (rad/sec)				
Filter	Method 1	Method 2	Method 3	Method 4	By Optimization
1	6.28	15.71	6.28	50.00	25.89
2	34.56	34.56	19.86	54.00	41.61
3	62.83	53.41	62.83	58.00	56.01

Table 2. Parameter Choices for Various Methods

For purposes of illustration, three filters are used to construct an MMFA for this example problem. The values of ω_n chosen using each of the four methods above are shown in Table 2. The weighting matrix $\mathbf{W}$ is chosen as the identity matrix, $\mathbf{I}$, and the truth model is the same dimension as the filter models ($\boldsymbol{T} = \mathbf{I}$).

Also shown in Table 2 are the values of ω_n chosen by the numerical optimization described in Section 2.1.2. For this optimization, it is assumed that the filters implemented are full state, constant gain filters. The gain used in the filters is the steady state Kalman filter gain assuming the correct values of $\boldsymbol{Q}_d$ and R.

Methods one through three are geometric in the parameter space and require no further discussion. Method four requires a trial and error approach, calculating the steady state autocorrelation of $\tilde{\mathbf{x}}(t_i^+)$ as a function of $\mathbf{a}$ for a "nominal" filter and using a predesignated increase in the error autocorrelation (e.g. 45%) to signal the point at which another filter is required. It was found that $\omega_n = 50.00$ is a good nominal value for this problem. At values of $\omega_n < 50.00$, the error autocorrelation never exceeds 145% of the value at $\omega_n = 50.00$. As shown in Figure 14, for $\omega_n > 50.00$, the value of the error autocorrelation reaches 145% of the nominal value around $\omega_n = 54.00$. Taking $\omega_n = 54.00$ as the nominal, the value of the error autocorrelation reaches 145% of this nominal around $\omega_n = 58.00$, as is shown in Figure 15.

3.1.3 Preliminary Analysis. A preliminary analysis is accomplished by calculating the cost functional, J^2 of Equation(49), for each choice of parameters. Plots of the steady state prediction error autocorrelation are shown in Figures 16 - 20 as a function of the truth system parameter for each of the design methods. The optimal prediction error autocorrelation, which would require a filter at every value of **a**, is also shown to illustrate how the estimation degrades when the true parameter is far from the parameters assumed within the filters. Triangles on the frequency axis denote the filter parameter values.

The discontinuities in the curves show the points in the parameter space where the MMFA switches filters. It is interesting to note that, as the true parameter increases, the MMFA switches to a filter with higher prediction error autocorrelation. Since our objective is to find a system with minimum estimation error, this is to our detriment. A maximum entropy with unit covariance (ME/I) estimator [25] is one step closer to our objective, as the filter selected in the ME/I estimator is the one with the lowest residual autocorrelation. This is discussed further in Section 3.1.5.2.

The cost functionals are calculated by integrating the plots in Figures 16 - 20, and these are tabulated in Table 3 for each of the design methods. To compare the preliminary results, we use method 1 as the baseline and make relative comparisons. It is evident from Figure 16 that placing the filters by method 1 puts the most

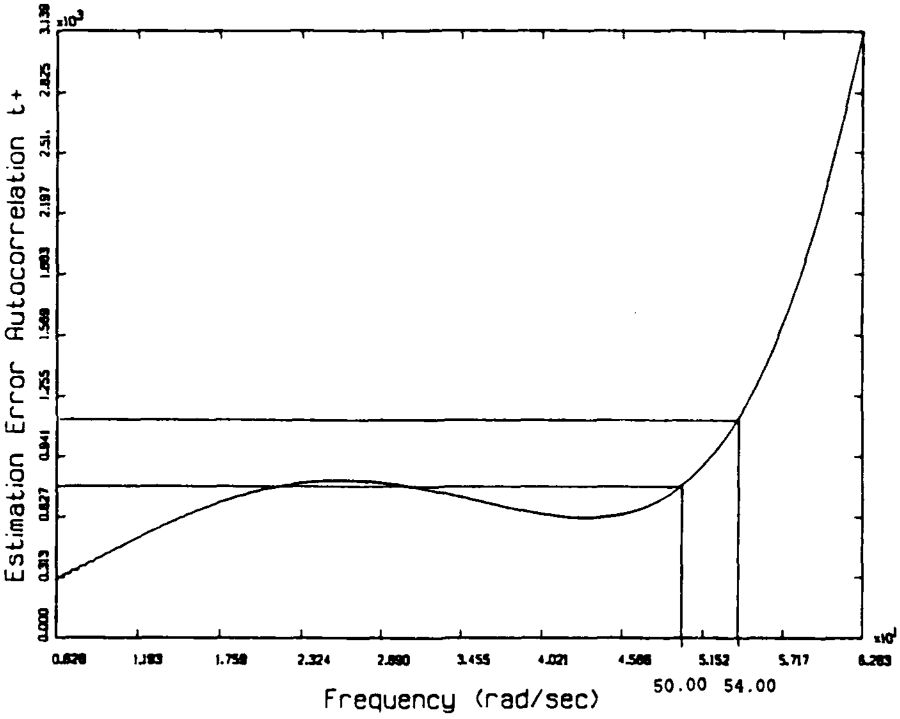


Figure 14. $\tilde{\mathbf{x}}(t_i^+)$ Error Autocorrelation for Method 4, $\omega_{\text{nom}} = 50.00$

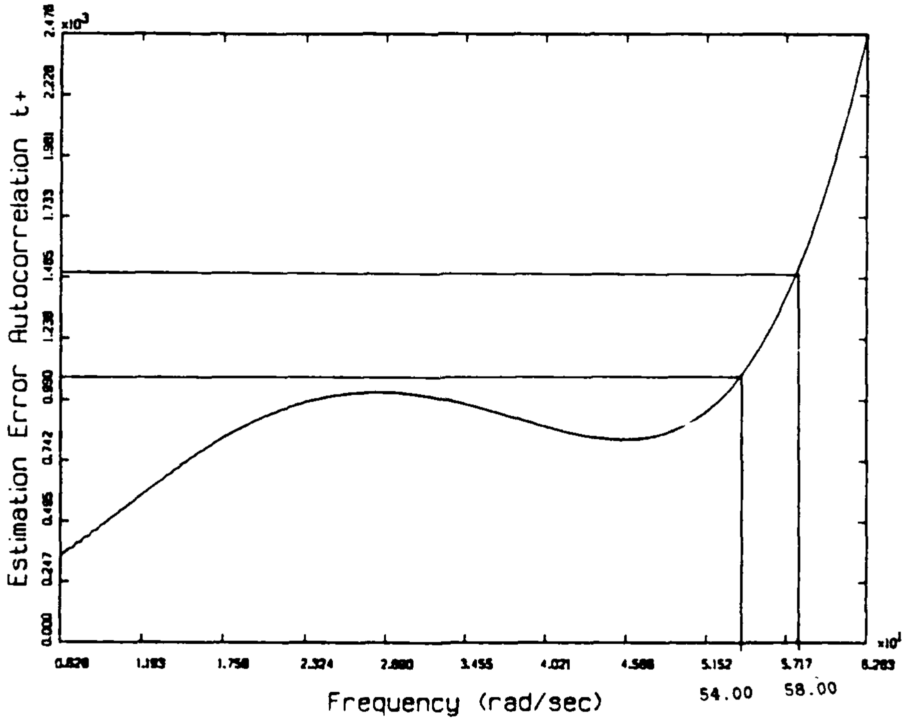


Figure 15. $\tilde{\mathbf{x}}(t_i^+)$ Error Autocorrelation for Method 4, $\omega_{\text{nom}} = 54.00$

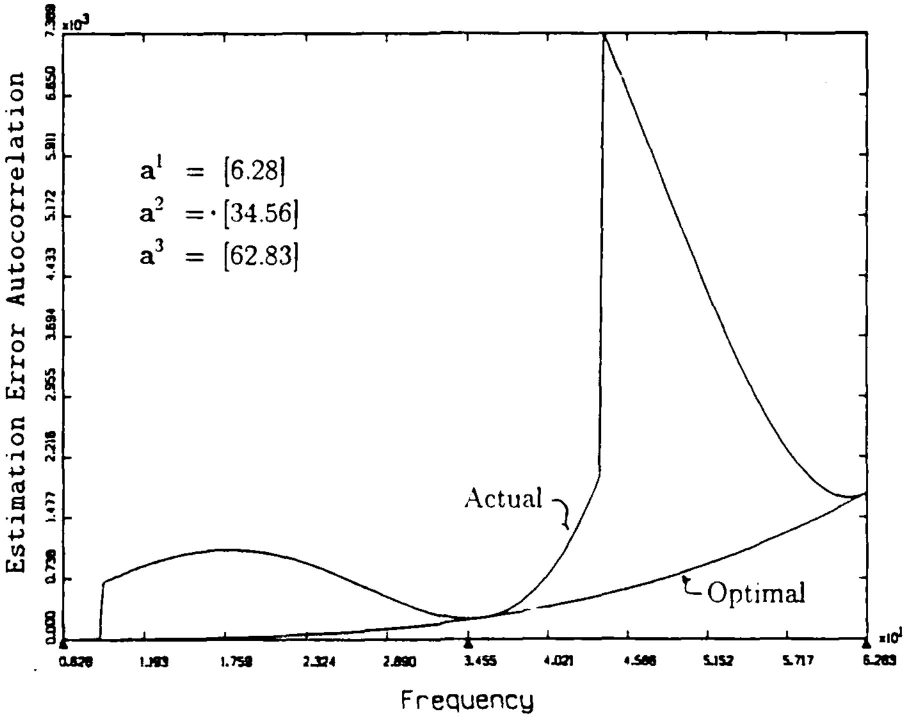


Figure 16. $\tilde{\mathbf{x}}(t_i^-)$ Autocorrelation Using Method 1

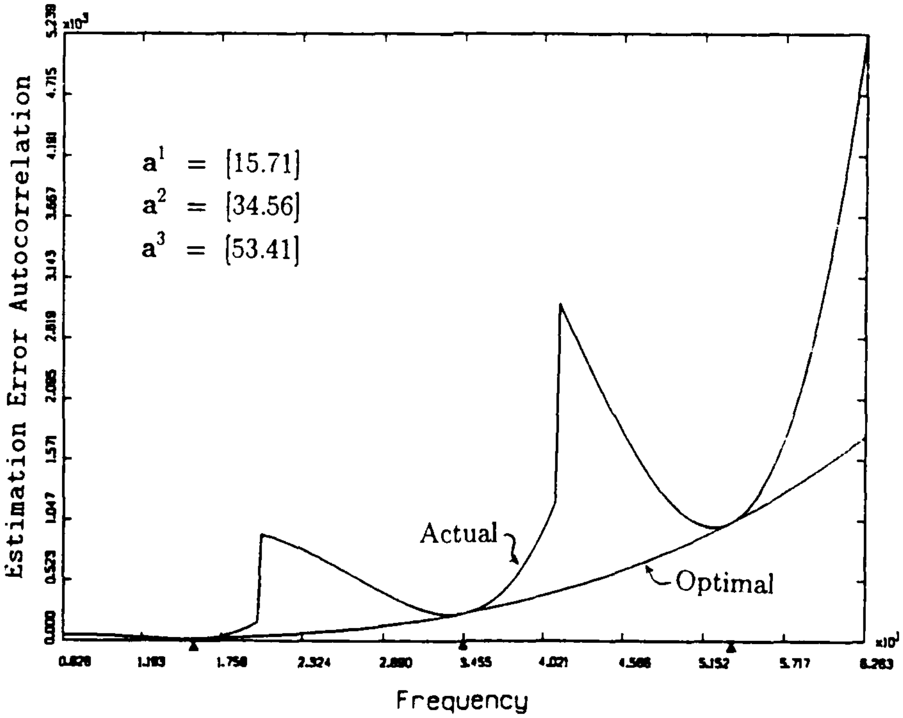


Figure 17. $\tilde{\mathbf{x}}(t_i^-)$ Autocorrelation Using Method 2

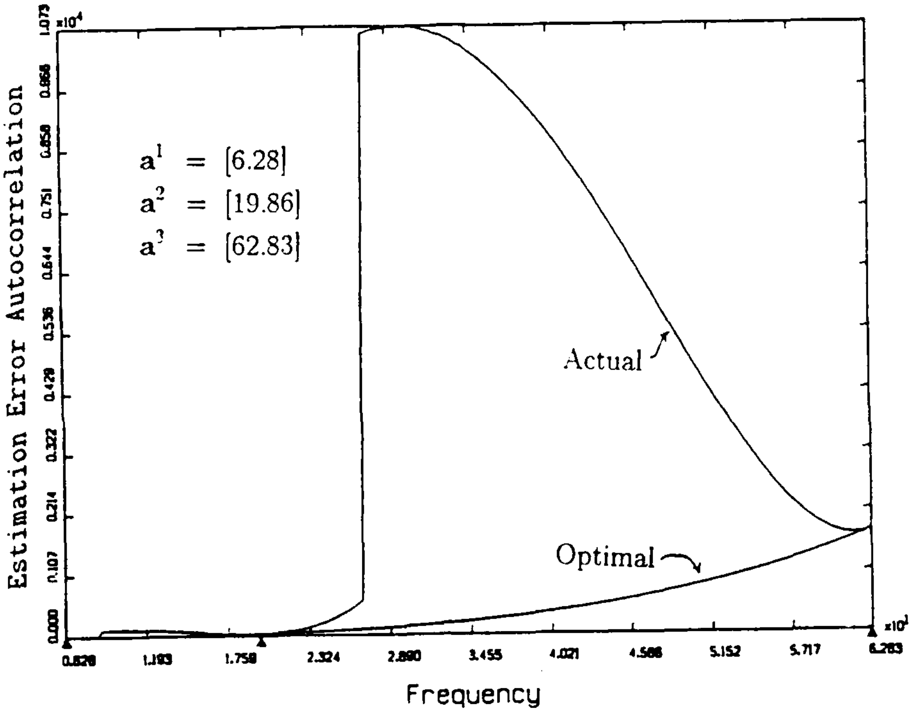


Figure 18. $\tilde{\mathbf{x}}(t_i^-)$ Autocorrelation Using Method 3

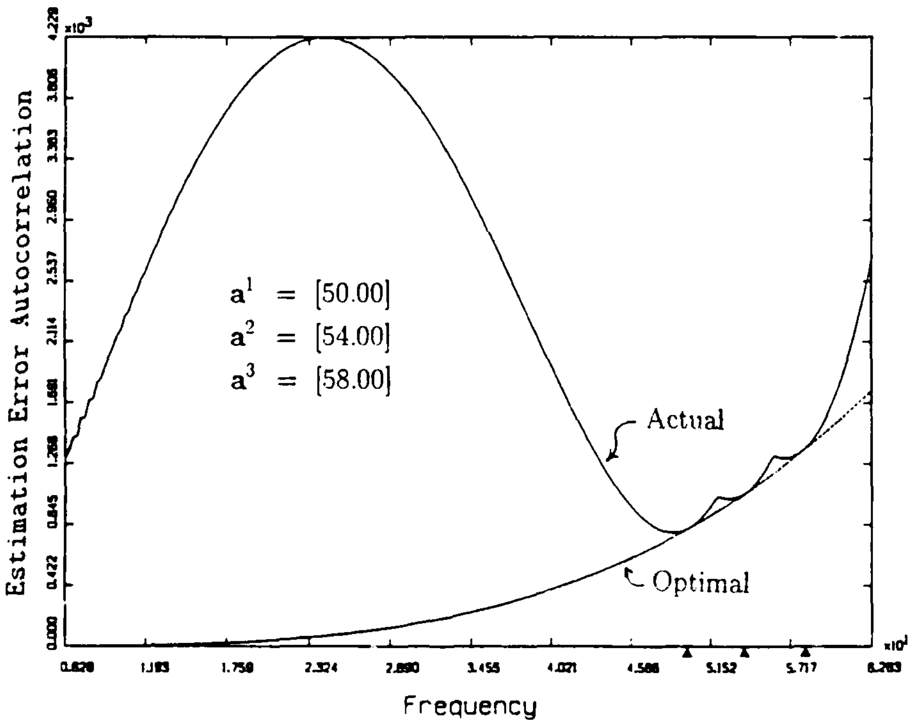


Figure 19. $\tilde{\mathbf{x}}(t_i^-)$ Autocorrelation Using Method 4

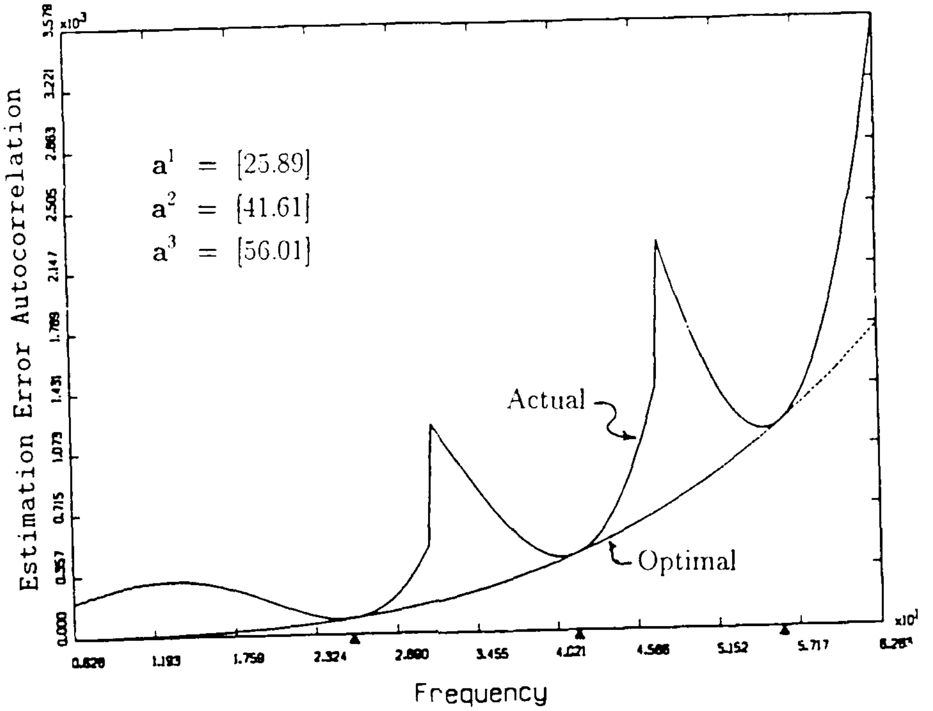


Figure 20. $\tilde{\mathbf{x}}(t_i^-)$ Autocorrelation Using Basic Optimization Method

Design Method	J^2		
	(m^2)		(m^2/s^2)
Method 1	0.1408	+	1876
Method 2	0.0845	+	986
Method 3	0.2692	+	4283
Method 4	0.1810	+	2482
Optimization	0.0742	+	797

Table 3. Cost Functional Evaluation for Various Methods

parameter space between filters. Since the estimation degrades as the true parameter moves away from a representative parameter, it seems reasonable to limit the space between filters. This thinking inspired method 2, and as shown in Figure 17, this method reduces the degradation in estimation error. As shown in Table 3, going from method 1 to method 2 cuts the cost functional in half. For this problem, method 2 is the best of the geometric methods.

The results for method 3 illustrate how improper reasoning can hurt a design. It seemed reasonable at the time for Hentz [19, 39] to choose a logarithmic division of the frequency range for his design. As shown in Table 3, this provides a cost functional over four times that of method 2. It is evident from these figures that it is better to place the filters closer together at higher frequencies.

Method 4 was developed in an attempt to place the filters on the basis of estimator performance instead of geometrically in the parameter space. One problem with this method is that the filter selection is not taken into account. It is evident from Figure 19 that the MMFA switches filters somewhere between the two representative parameters so that the 45% increase in error autocorrelation is not experienced between the filters, but it is reached on the ends of the region. Even if the switching were taken into account, this design is based on the error at t_i^+, and the filter selection is based upon the prediction error (at t_i^-). This would require calculating error autocorrelation both at t_i^- and t_i^+. It would be more reasonable to use this method with data at t_i^- as it shows more dependency on model adequacy. Method 4 is the most difficult by far, as it involves a great deal of trial and error to get the parameter region covered with a prespecified number of filters. Both the nominal parameter value and the amount of degradation allowed between filter parameters are chosen by the designer.

The set of parameters found by the optimization algorithm produces a system with a cost functional around 19.2% lower than the nearest competitor. This would lead us to believe that an MMFA implemented with the optimal discretization

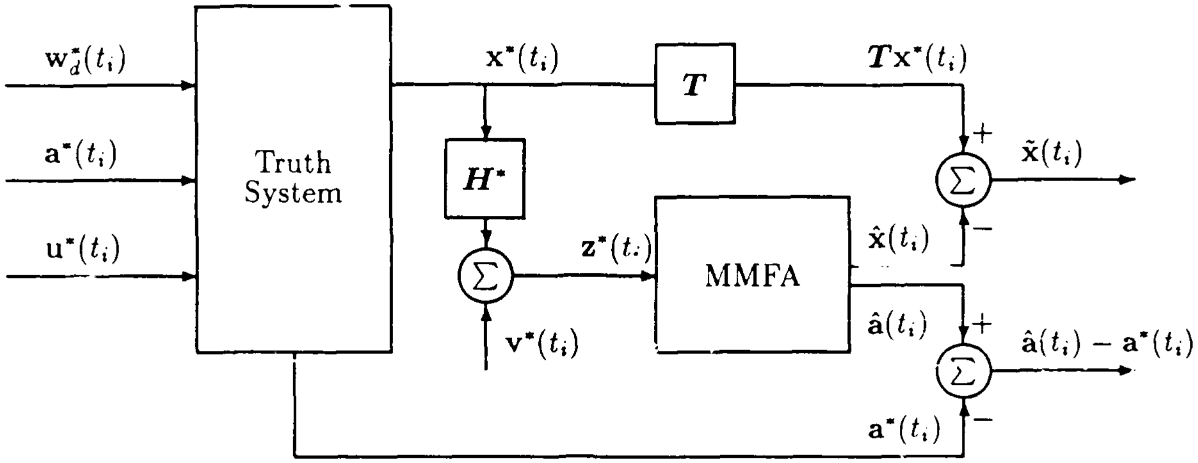


Figure 21. MMAE Simulation Block Diagram

would provide an estimate with the lowest mean square prediction error on average. Moreover, the optimized system should exhibit mean square prediction errors approximately 19.2% lower than those exhibited by the method 2 design. To test this conjecture, a simulation of the different estimators is in order.

3.1.4 Estimator Simulation. To evaluate the parameter discretization found by the optimization algorithm described in Chapter II, a Monte Carlo simulation was performed. A block diagram of the computer simulation is shown in Figure 21. Equation (127) was used to simulate the truth system with no input ($u^*(t_i) = 0$), and it was driven by a simulated discrete time white noise process that was the same for each estimator. The truth model was the same structure as the filter models, and the only difference was the value of ω_n, so the transformation matrix $\boldsymbol{T}$ was the identity matrix. Each MMFA was tested by running 20 simulations at each of 200 values of ω_n (a^*). Each simulation was ten seconds in duration and ω_n remained constant throughout the simulation. Equation (1) was used to propagate the probabilities forward from an a priori probability distribution of $p^1(t_0) = p^2(t_0) = p^3(t_0) = 1/3$, and there was no lower bound on the probabilities.

Quantity of Interest	Design Method				
	Meth. 1	Meth. 2	Meth. 3	Meth. 4	Opt.
MS Position Prediction Error (meters^2)	0.1348	0.0837	0.2590	0.1756	0.0731
MS Velocity Prediction Error (meters^2/sec^2)	1792	982	4101	2371	794
MS Parameter Estimation Error (rad^2/sec^2)	138.8	39.4	265.3	495.8	59.6

Table 4. Monte Carlo Simulation $\tilde{\mathbf{x}}(t_i^-)$ Results

Quantity of Interest	Design Method				
	Meth. 1	Meth. 2	Meth. 3	Meth. 4	Opt.
Mean Square Position Error (meters^2)	0.01003	0.00899	0.01033	0.01433	0.00911
Mean Square Velocity Error (meters^2/sec^2)	439.7	315.0	719.0	560.0	285.5

Table 5. Monte Carlo Simulation $\tilde{\mathbf{x}}(t_i^+)$ Results

3.1.5 Results. The results of the Monte Carlo simulations are tabulated in Tables 4 and 5. Table 4 presents the average mean square prediction error at t_i^- for each simulation calculated by

$$\begin{array}{c}\text{Mean Square Position}\\ \text{Prediction Error}\end{array} = \frac{1}{4000}\sum_{\text{runs}=1}^{4000}\{\frac{1}{800}\sum_{i=201}^{1000}[\hat{x}_1(t_i^-) - x_1^*(t_i)]^2\} \quad (130)$$

$$\begin{array}{c}\text{Mean Square Velocity}\\ \text{Prediction Error}\end{array} = \frac{1}{4000}\sum_{\text{runs}=1}^{4000}\{\frac{1}{800}\sum_{i=201}^{1000}[\hat{x}_2(t_i^-) - x_2^*(t_i)]^2\} \quad (131)$$

$$\begin{array}{c}\text{Mean Square Parameter}\\ \text{Estimation Error}\end{array} = \frac{1}{4000}\sum_{\text{runs}=1}^{4000}\{\frac{1}{800}\sum_{i=201}^{1000}[\hat{a}(t_i^+) - a^*(t_i)]^2\} \quad (132)$$

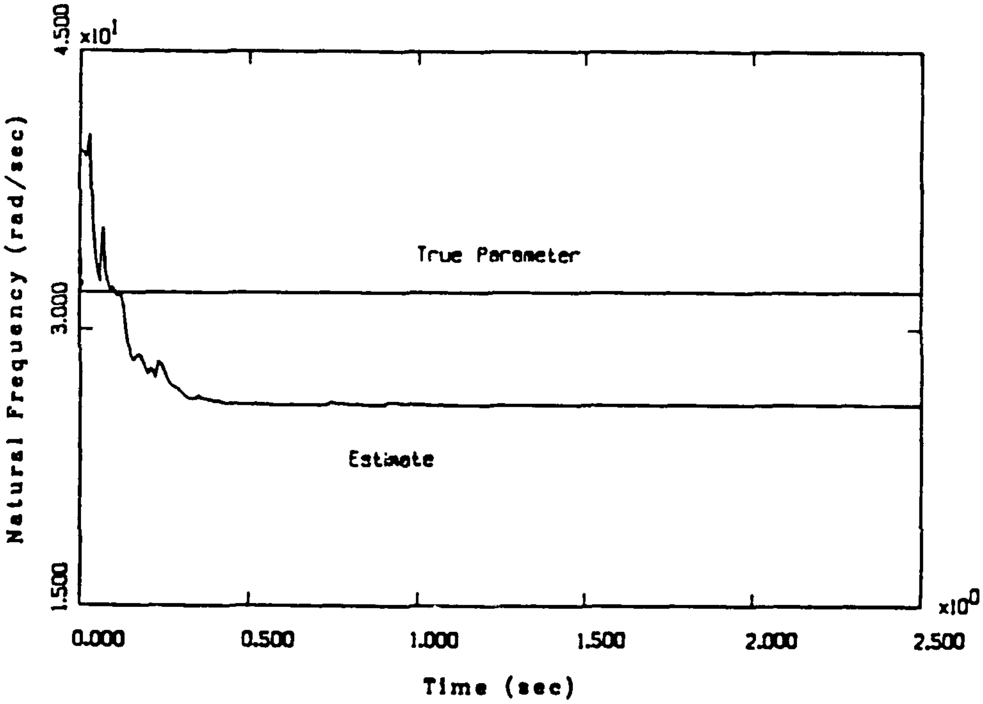


Figure 22. Parameter Estimation Convergence

Table 5 presents the average mean square estimation error at t_i^+ for each simulation calculated by

$$\begin{matrix}\text{Mean Square}\\ \text{Position Error}\end{matrix} = \frac{1}{4000}\sum_{runs=1}^{4000}\{\frac{1}{800}\sum_{i=201}^{1000}[\hat{x}_1(t_i^+) - x_1^*(t_i)]^2\} \qquad (133)$$

$$\begin{matrix}\text{Mean Square}\\ \text{Velocity Error}\end{matrix} = \frac{1}{4000}\sum_{runs=1}^{4000}\{\frac{1}{800}\sum_{i=201}^{1000}[\hat{x}_2(t_i^+) - x_2^*(t_i)]^2\} \qquad (134)$$

The simulation results are averages from 2 seconds through 10 seconds. This was done to eliminate the effects of the initial identification transient from the results. Figure 22 is a plot of the parameter estimate versus the true parameter for a representative sample run from the Monte Carlo simulation. The estimator converges to the selected filter with $p^{sel} = 1$ within the 2 second window, and the results tabulated above accurately reflect the steady state mean square error. In a system with a varying parameter, the probabilities must be bounded below, as discussed in Section 2.1.3.3, to prevent the adaptation mechanism from experiencing lockup. This lower bound is discussed further in Section 3.1.5.4.

The optimized estimator exhibits average mean square position and velocity prediction errors lower than all other methods, while providing good parameter estimation. This indicates the benefits of the proposed discretization strategy in discerning the distinctions between alternative models. The benefits are clearly seen when the results are compared to method 4, which is the method currently used at AFIT. The optimal estimator produces a mean square prediction error approximately one-third of that produced by the method 4 system. As expected, the estimator with the lowest overall prediction error is also the estimator with the lowest overall $\hat{x}(t_i^+)$ error.

It was expected that the filter designed through optimization would exhibit a mean square error 19.2% lower than that of method 2. The simulation results show that the optimized MMFA has a 19.1% lower mean square error. Moreover, the values of error calculated in the cost functional computations are a very accurate prediction of the simulation results. Figures 23 and 24 are plots of the mean square prediction error ($\pm 1\sigma$) shown in solid lines from the simulation of the optimized design. The predicted mean square prediction error from the cost functional calculations is shown in dotted lines. The simulation results in a mean square position prediction error 0.73% higher on average than was predicted. The mean square velocity prediction error is 0.70% lower on average than was predicted.

Note that the standard deviation, σ, is small when the true parameter value is close to a parameter in the representative parameter set, and grows larger as the true parameter value moves away from the representative parameters. This is indicative of the degradation in performance of an estimator based on an incorrect parameter.

The basic optimization algorithm successfully produced a representative parameter set for an MMFA with a prediction error lower than any previously accepted method. The numbers are specific to this example, and different amounts of benefit are likely for other problems. This example shows that the basic design method is a valuable tool, both for predicting the performance of MMFA designs, and for

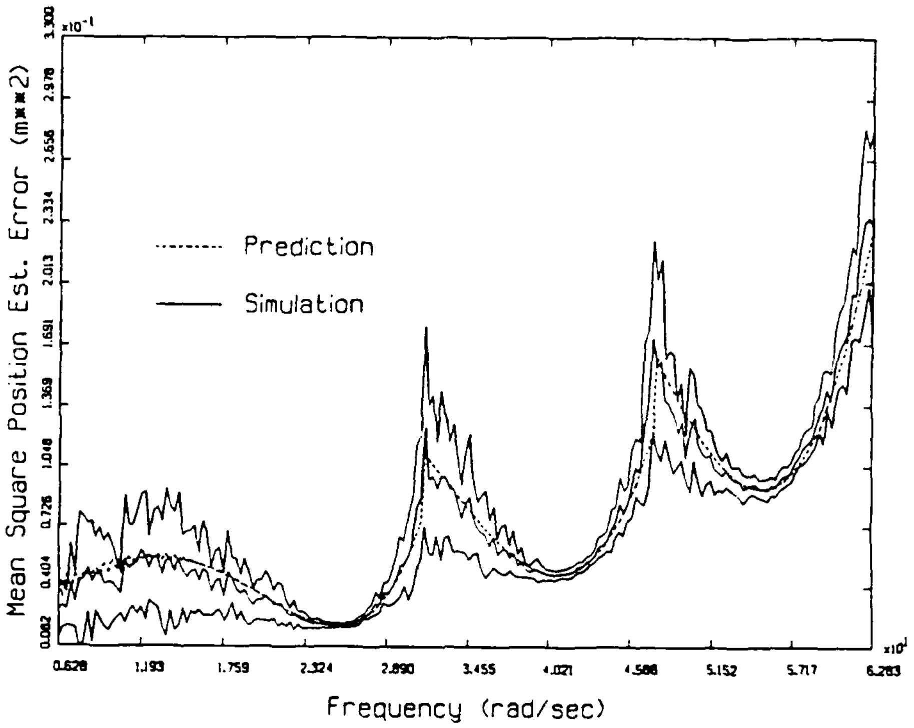


Figure 23. Predicted Versus Simulation Position Prediction Error

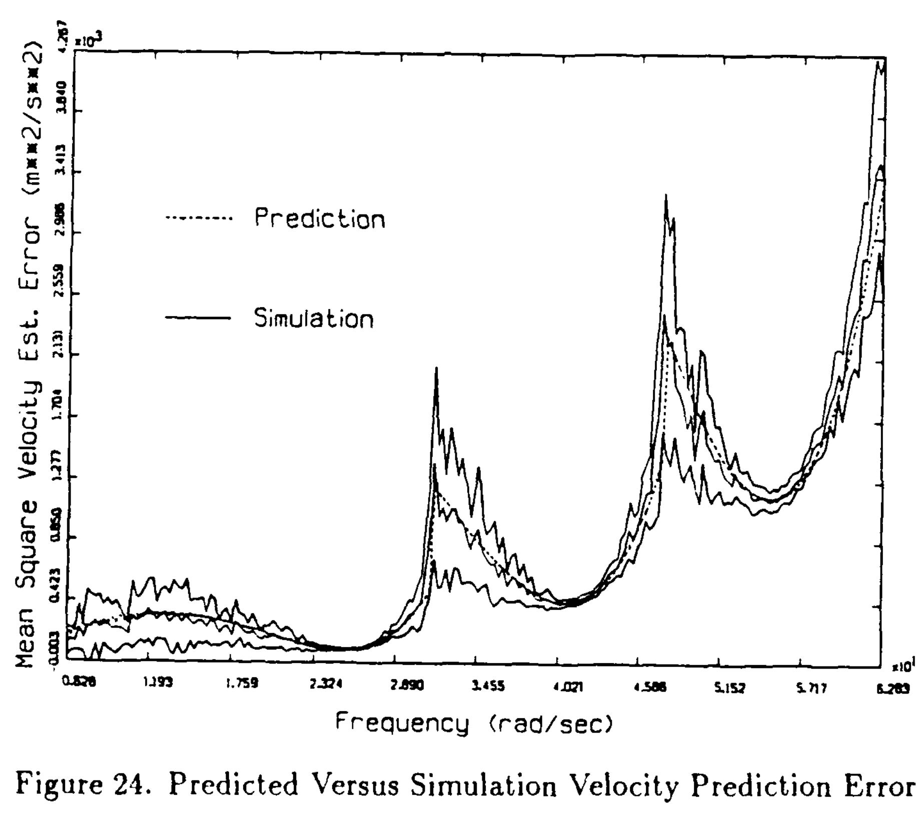


Figure 24. Predicted Versus Simulation Velocity Prediction Error

designing MMFAs which provide better estimation performance. The next few sections illustrate how the basic algorithm can be modified.

3.1.5.1 Effect of Various Weighting Matrices. The weighting matrix $\mathbf{W}$ was included in the development to provide a means for tradeoff in systems where minimizing the estimation error for one state has a detrimental effect on the estimation accuracy of another state. Such conflict is common in practical engineering design. Our chosen example problem exhibits a rather interesting phenomenon. Chosing the weighting matrices

$$\mathbf{W}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

and

$$\mathbf{W}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

produce the same parameter discretization, namely

$$\begin{aligned} \mathbf{a}^1 &= [25.89] \\ \mathbf{a}^2 &= [41.61] \\ \mathbf{a}^3 &= [56.01] \end{aligned}$$

which is the same parameter set found in Section 3.1.2. It seems that, for this example, the position prediction error is minimized when the velocity prediction error is minimized.

3.1.5.2 Using The ME/I Hypothesis. The large jumps in the true state estimation error autocorrelation, seen in Figures 16 - 20, inspired a search for an MMFA that selects the filter with the lowest estimation error autocorrelation. The ME/I mechanization, described in Section 2.1.3.1, causes the MMFA to choose the filter with the lowest residual autocorrelation, $\mathrm{E}\{\mathbf{r}^{k\mathsf{T}}(t_i)\mathbf{r}^k(t_i)\}$, per Equation (2).

This has the effect of shifting the switch point as illustrated in Figures 25 and 26. Figure 25 shows the residual prediction error autocorrelation of the basic optimized estimator of Section 3.1.3 assuming the standard ME/M hypothesis and the ME/I hypothesis. The estir^ator using the ME/I hypothesis selects the filter with the lowest residual error autocorrelation. Figure 26 is the corresponding velocity prediction error autocorrelation.

The cost functionals for the five systems described in Section 3.1.2 are tabulated in Table 6. In most cases the costs are significantly smaller than the corresponding costs for the basic system shown in Table 3. This would lead us to believe that an MMFA with the ME/I hypothesis would perform better. By replacing Equation (54) in the optimization algorithm with Equation (58), the optimization algorithm developed in Section 2.1.2 can be used to optimize such a system. For the example problem, the algorithm finds the representative parameter set

$$\begin{aligned} \mathbf{a}^1 &= [26.91] \\ \mathbf{a}^2 &= [43.50] \\ \mathbf{a}^3 &= [57.00] \end{aligned}$$

The cost for this system (see Table 6) combines a slightly higher position prediction error with a lower velocity prediction error to produce an overall cost which is slightly lower than the cost of the system found in the basic optimization. This set is shifted to the right slightly from the basic optimized solution, reflecting the lower predicted error at intermediate points exhibited in Figures 25 and 26.

This system was evaluated by running the same type of Monte Carlo simulation as described in Section 3.1.4, but using the ME/I hypothesis. The hypothesis conditional probability calculation was changed to use Equation (57) to calculate the conditional density. The results, tabulated in Tables 7 and 8, illustrate that, for each estimator, the ME/I mechanization provides a lower prediction error autocorrelation on average.

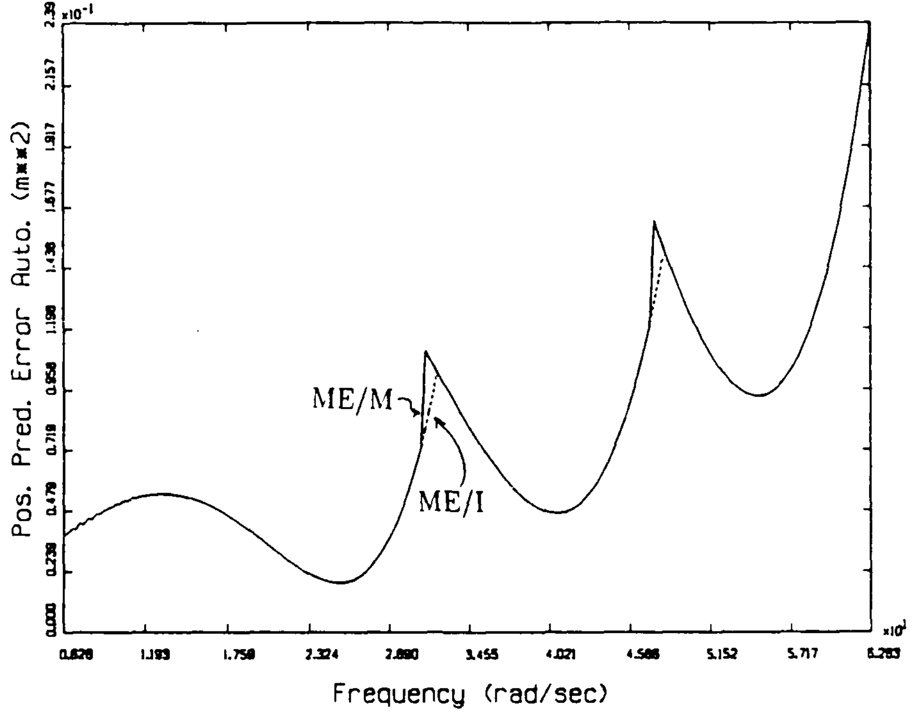


Figure 25. Residual Prediction Error Autocorrelation Comparison

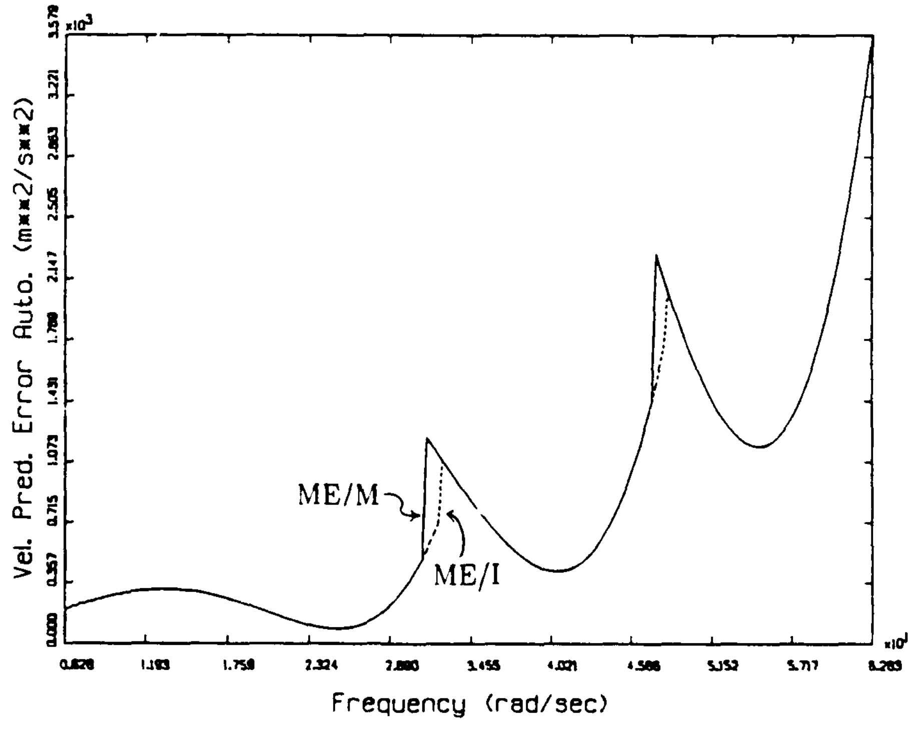


Figure 26. Velocity Prediction Error Autocorrelation Comparison

Design Method	$J^2_{\mathrm{ME/I}}$		
	(m^2)		(m^2/s^2)
Method 1	0.1297	+	1579
Method 2	0.0831	+	942
Method 3	0.2353	+	3274
Method 4	0.1839	+	2497
Basic Opt.	0.0739	+	783
ME/I Opt.	0.0743	+	777

Table 6. Costs for ME/I Mechanization

Quantity of Interest	Design Method					
	Meth. 1	Meth. 2	Meth. 3	Meth. 4	Basic Opt.	ME/I Opt.
MS Position Pred. Error (meters^2)	0.1240	0.0805	0.2277	0.1758	0.0702	0.0710
MS Velocity Pred. Error (meters^2/sec^2)	1520	920	3163	2375	756	746
MS Para. Est. Error (rad^2/sec^2)	111.5	32.9	169.5	502.8	58.3	65.2

Table 7. ME/I Hypothesis Simulation $\tilde{\mathbf{x}}(t_i^-)$ Results

Quantity of Interest	Design Method					
	Meth. 1	Meth. 2	Meth. 3	Meth. 4	Basic Opt.	ME/I Opt.
MS Position Error (meters^2)	0.01075	0.00908	0.01574	0.01026	0.00902	0.00910
MS Velocity Error (meters^2/sec^2)	420 7	306.3	648.9	560.0	280.0	280.1

Table 8. ME/I Hypothesis Simulation $\tilde{\mathbf{x}}(t_i^+)$ Results

λ	$\mathbf{a}^1$	$\mathbf{a}^2$	$\mathbf{a}^3$	J^2		J^∞		$J = \lambda J^2 + (1-\lambda)J^\infty$	
				m^2	m^2/s^2	m^2	m^2/s^2	m^2	m^2/s^2
1.0	25.89	41.61	58.01	0.0742	797.1	0.2394	3575	0.0742	797.1
0.7	28.53	44.35	58.40	0.0760	805.3	0.1746	2533	0.1056	1324
0.3	28.53	46.50	58.98	0.0778	827.2	0.1633	2348	0.1377	1892
0.1	31.28	47.06	59.27	0.0799	842.5	0.1583	2265	0.1505	2123

Table 9. Optimal Parameter Sets for Hybrid Cost Functional

The simulation results for the optimized ME/I system at t_i^- support the prediction made in the preliminary analysis. The position prediction error is slightly higher than that for the system found with the basic optimization, whereas the velocity prediction error is lower than that for the system found with the basic optimization. These values combined produce a lower overall result. The results at t_i^+ show that the ME/I optimized system has a higher overall error, but the difference is on the order of the accuracy of the simulation; hence no conclusion can be drawn from these results.

The ME/I hypothesis is an ad hoc modification to the basic algorithm. The benefit to the example problem, though small, is measurable and could possibly be large in a system where $\boldsymbol{M}$ varies a great deal across the parameter set.

3.1.5.3 Adding a Maximum-Type Cost. In Section 2.1.3.2, the following hybrid cost functional was proposed in order to address maximum mean square error specifications:

$$J = \lambda J^2 + (1-\lambda)J^\infty \tag{135}$$

where λ is allowed to vary between a small ϵ and 1.

Table 9 is a summary of the results of optimizing with this hybrid cost functional using different values for λ. With λ equal to one, the optimization is the basic optimization problem with J^2. As λ is decreased, more emphasis is placed on the maximum estimation error autocorrelation, forcing the largest representative

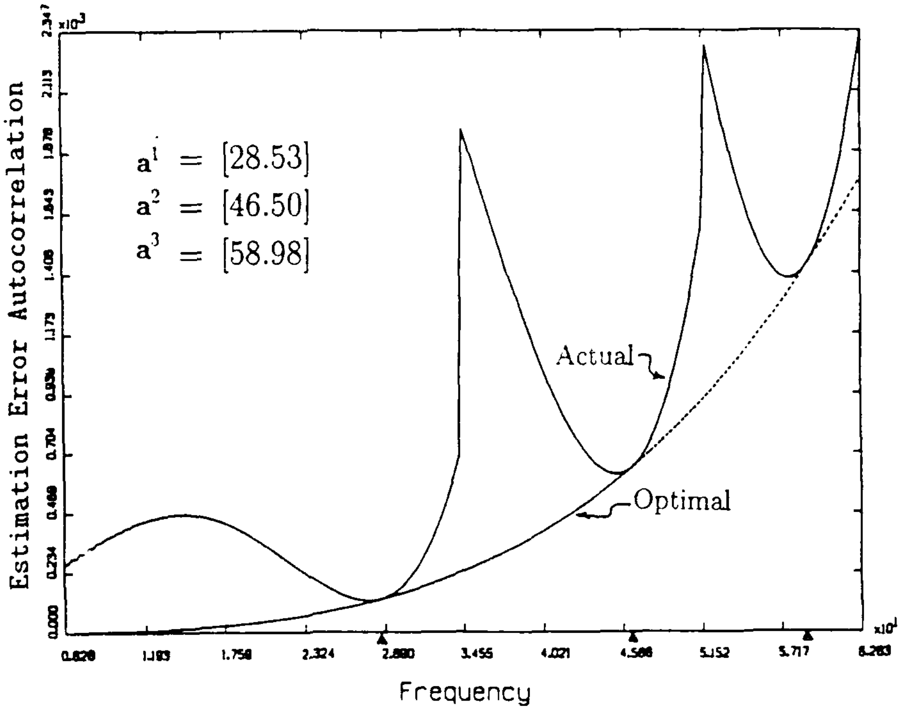


Figure 27. Prediction Error Autocorrelation for $\lambda = 0.3$

parameter value to increase (see Figure 27 as compared to Figure 20). This lowers the maximum estimation error autocorrelation (i.e., peak value of the curve) at the expense of increasing the average estimation error autocorrelation (i.e., area under the curve). There is a limit to the benefit derived from adding a maximum-type cost. The value of J^{∞} will never go below $0.1294\text{m}^2 + 1776\text{m}^2/\text{s}^2$, which is the value achieved by the true optimal solution which requires an uncountably infinite bank of filters.

3.1.5.4 Effect of Estimator Lower Bounds. We assumed, in the previous sections of this chapter, that the probability of the selected filter converges to one. Once this happens, the probabilities of the other filters are zero and the probability calculation will experience lockup, as explained Section 2.1.3.3. This is why the true natural frequency was constant in the simulations of Section 3.1.4.

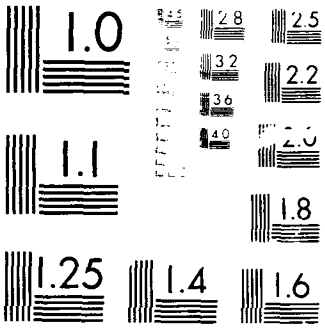
Using the equations developed in Section 2.1.3.3, it is possible to calculate the cost of an MMFA for any given lower bound on the probabilities. Table 10 shows

Probability Lower Bound	J^2	
	m^2	m^2/s^2
0.000	0.07416	797.1
0.001	0.07419	797.5
0.010	0.07465	803.0
0.050	0.08172	876.0
0.100	0.10212	1079.0

Table 10. Cost Variation with Different Lower Bounds on the Probabilities

Quantity of Interest	p_{min}			
	0.000	0.001	0.010	0.100
MS Position Prediction Error (meters^2)	0.07314	0.06812	0.06470	0.08325
MS Velocity Prediction Error (meters^2/sec^2)	794.4	744.9	718.3	938.6
MS Parameter Estimation Error (rad^2/sec^2)	59.6	58.8	62.1	105.7
MS Position Error (meters^2)	0.00899	0.00897	0.00900	0.01068
MS Velocity Error (meters^2/sec^2)	285.7	283.5	287.9	382.7

Table 11. Simulation Results with Various Values of p_{min}


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

how the cost of the basic optimized solution varies with a change in the probability lower bound. As $p_{\min}$ is increased, the cost increases as expected.

The results of simulations with various values of $p_{\min}$ are shown in Table 11. It is evident from these results that adding a lower bound to the probabilities has an additional benefit which cannot be predicted from the cost calculations. It can be explained, however, with the help of Figures 28 and 29 which are plots of the parameter estimate in a sample simulation with $p_{\min} = 0.001$ and $p_{\min} = 0.010$.

In the basic simulations, the MMFA converges to one of the filters in the bank and stays there. Since there are a finite number of filters and an uncountably infinite number of possible true parameters, the selected filter in the MMFA is almost surely based upon the wrong parameter. Bounding the probabilities below allows the MMFA to move off the wrong filter if the dynamics in the system match a different filter at some point during the simulation. The most benefit is derived at the points in the parameter space where the selection switches filters. Figures 30 and 31 are plots of Monte Carlo simulation results with $p_{\min} = 0.001$ using the basic optimization design. Note how the mean square error from the simulation is lower around the switch points of the MMFA, as compared to the results displayed in Figures 23 and 24. This benefit is large and masks the detrimental effect computed above for small values of $p_{\min}$. However, for larger values of $p_{\min}$, the detrimental effect becomes dominant as illustrated by the results in Table 11. The amount of benefit gained is likely to be problem dependent.

3.1.6 Computational Burden. The computations required for this research were accomplished on an Elxsi computer with 10 central processing units (CPUs). Each CPU is rated at 8 VAX MIPS, but the overhead due to the multiple user configuration effectively lowers this rating. The cost functional calculations for the example problem takes approximately six minutes of CPU time with the parameter set divided into 200 intervals. The majority of the computation time is spent on

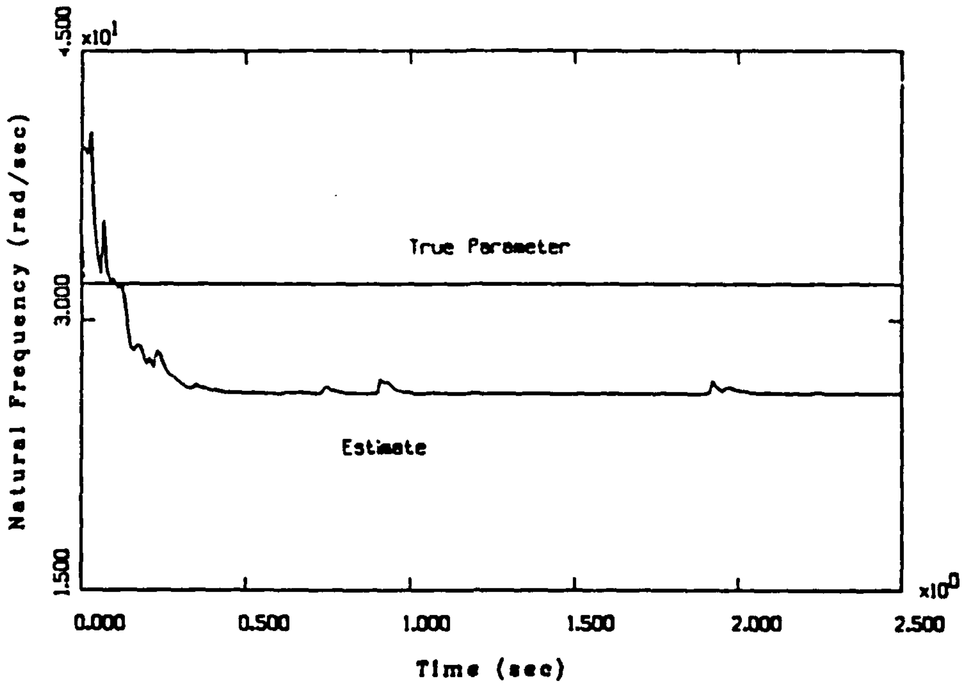


Figure 28. Estimator Convergence with $p_{min} = 0.001$

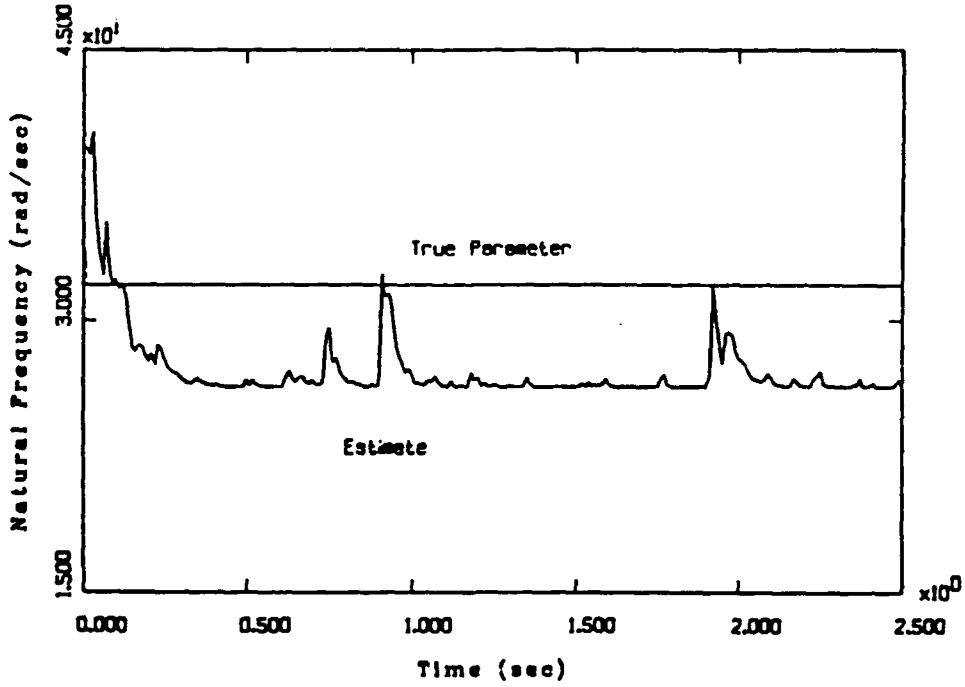


Figure 29. Estimator Convergence with $p_{min} = 0.010$

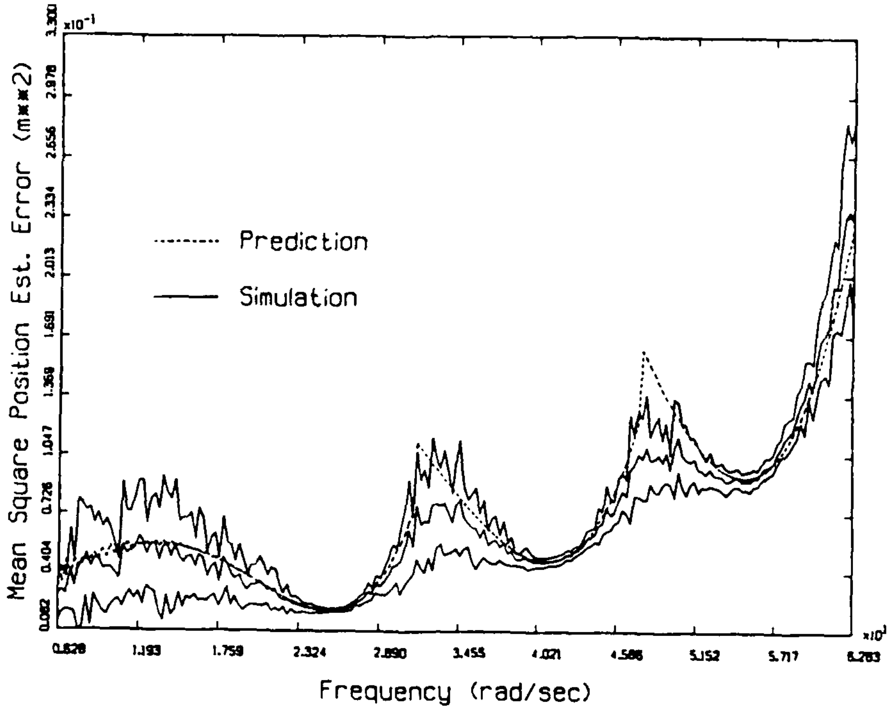


Figure 30. Predicted Versus Simulation Position Prediction Error $p_{\min} = 0.001$

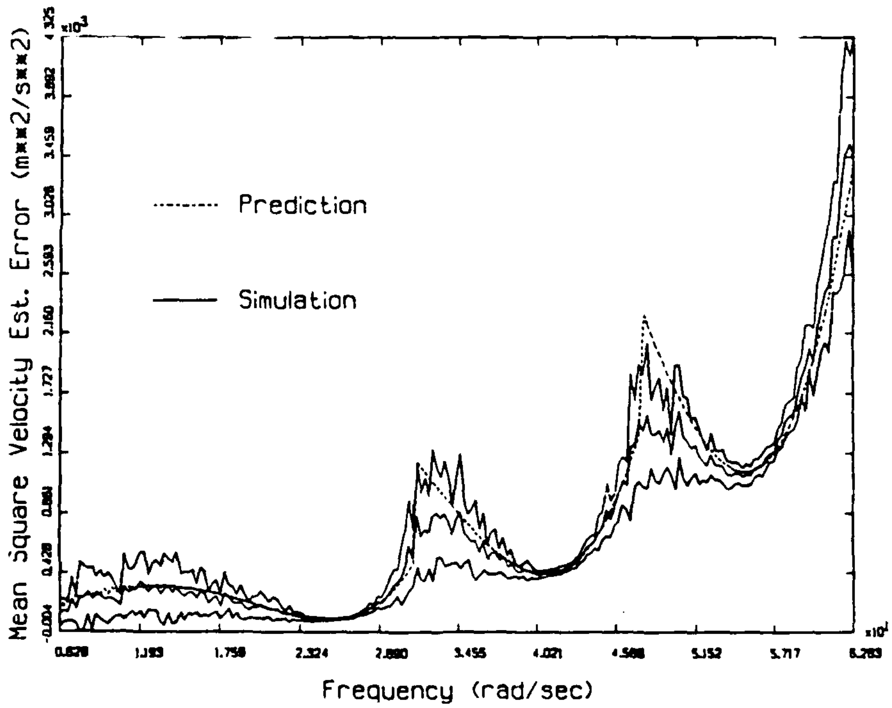


Figure 31. Predicted Versus Simulation Velocity Prediction Error $p_{\min} = 0.001$

solving the Lyapunov equation, Equation (47), for each of the three filters, at each point evaluated on the curve. The software used in this research solves the Lyapunov equations by computing Equation (47) for a sufficient number of steps to reach the desired level of accuracy. A good Lyapunov equation solving routine would make a dramatic impact on the amount of computing power necessary for any problem. With a lower bound on the probabilities, the problem changes from solving three four-dimensional Lyapunov equations to solving one eight-dimensional Lyapunov equation at each point evaluated, and the cost functional calculation takes approximately 12 minutes of CPU time.

The estimator optimization algorithm is more computationally intensive, as it involves calculating the cost functional at each step in the minimization. Equation (47) must be solved for each filter, at each point evaluated on the curve, for each cost functional evaluation during the optimization. As an example, if 200 points are evaluated on the curve for each cost functional evaluation, and the optimization algorithm takes 40 evaluations, $3 \times 200 \times 40 = 24,000$ Lyapunov equations must be solved. A representative optimization for the basic example problem takes approximately 3.5 hours of CPU time.

3.1.7 Summary. Using a vector minimization routine and an appropriate cost functional, a *multiple model filtering algorithm* (MMFA) was designed which provides the minimum mean square prediction error (at t_i^-), when compared with other previously accepted design methods. This is important because many control strategies depend upon the ability to provide good prediction of future behavior of the plant before the next sampled data measurement. As anticipated, this system also provides lower mean square estimation error (at t_i^+) than any previously accepted method.

This system was shown in simulation to provide results for $\hat{\mathbf{x}}(t_i^-)$ very close to values predicted in the design method. This makes the cost functional computation a very useful analysis tool in as much as it accurately predicts the performance of the MMFA across the parameter space. Computing the cost functional, and specifically producing plots like Figures 16 through 20, provides a good measure of the quality of any given design. Based on this information, one can decide whether or not an optimization is warranted.

Replacing the ME/M hypothesis with the ME/I hypothesis allowed the MMFA to converge to the filter with the lowest residual autocorrelation. This resulted in better performance for all of the previously accepted designs. The optimization algorithm was modified to implement this hypothesis and produced a design different from the basic optimized design. This ME/I optimized design provides the lowest mean square prediction error, although no substantive benefit was seen in the $\hat{\mathbf{x}}(t_i^+)$ results.

It was shown in Section 3.1.5.3 that a maximum-type cost can be added to the original algorithm to find systems with a lower maximum error autocorrelation than the system found with the basic approach. This decrease in maximum mean square prediction error is found at the expense of increased average mean square prediction error.

The ad hoc method of placing a lower bound on the probabilities allows the system to adapt to changing parameters. The calculations required for the optimization algorithm can be augmented to take this into account, and they predict that the lower bound will have a detrimental effect on the estimation performance. In simulation, however, the lower bound is also shown to have a beneficial effect by allowing the MMFA to move off of a mismatched filter after the MMFA has converged to that filter. At low values of $p_{\min}$, this beneficial effect masks any detrimental effects.

3.2 Parameter Estimator Evaluation

It is important to note that the optimal state estimator constructed in Section 3.1.2 is not necessarily the best parameter estimator. The MMFA constructed

Design	J_a^2
Method	$(\text{rad}^2/\text{s}^2)$
Method 1	149.0
Method 2	40.1
Method 3	280.2
Method 4	494.8
Basic Opt.	58.6
Pai ι. Opt.	37.0

Table 12. Parameter Estimation Costs

using method 2 has the lowest mean square parameter estimation error of the five methods tested. This leads us to the conclusion that the best state estimator is not necessarily the best parameter estimator. By using the algorithm in Section 2.2.1, a parameter estimator is found which is optimal in the sense that it minimizes the cost functional of Equation (74). The set of representative parameters found for the example problem is:

$$\begin{aligned} \mathbf{a}^1 &= [18.51] \\ \mathbf{a}^2 &= [35.30] \\ \mathbf{a}^3 &= [53.09] \end{aligned}$$

This system is spread more evenly in the parameter space than the optimized state estimator. However, it is not quite the same as the method 2 design because its switch points are based on the state estimation performance instead of geometry in the parameter space. As shown in Table 12, a system designed with this parameter set has a lower cost than all other methods.

A three-filter MMFA based upon this parameter set was simulated in the same manner as the other estimators in Section 3.1.4. The results are tabulated in Table 13 along with the results from the previous tests. Note that the optimization produces a parameter estimator with the lowest mean square parameter estimation error. Also

Quantity of Interest	Design Method					
	Meth. 1	Meth. 2	Meth. 3	Meth. 4	Basic Opt.	Para. Opt.
MS Para. Est. Error (rad^2/sec^2)	138.8	39.4	265.3	495.8	59.6	37.5

Table 13. Parameter Estimation Results

note that there is good agreement between the predictions in Table 12 and the results in Table 13.

3.2.1 Effect of Various Weighting Matrices. The example problem used in this document has one variable, so the presence of a weighting matrix in the cost functional is inconsequential. However, the weighting matrix might be useful in other problems where it is desirable to weight one parameter more than another.

It may be necessary to use a weighting matrix to give parameters equal weight if they have different scales. Let's assume that the damping ratio was allowed to vary along with the natural frequency in the example problem. The damping ratio can vary in the interval [0, 1]. The natural frequency can vary in the interval [6.28 rad/sec, 62.83 rad/sec]. Using a weighting matrix of

$$\mathbf{W} = \begin{bmatrix} 1 & 0 \\ 0 & (56.55)^2 \end{bmatrix}$$

for

$$\mathbf{x} = \begin{bmatrix} \omega_n \\ \zeta \end{bmatrix}$$

would put the parameters on more equal terms in the minimization, in the sense that the same weight is given to equal percentage variations of the admissible parameter ranges.

Design	$J^2_{aME/I}$
Method	(rad^2/s^2)
Method 1	119.28
Method 2	33.96
Method 3	174.04
Method 4	493.45
Basic Para. Opt.	33.52
ME/I Para. Opt.	33.24

Table 14. Costs for Parameter Estimators Using the ME/I Mechanization

3.2.2 Using the ME/I Hypothesis in the Parameter Estimator. The ME/I hypothesis, as described in Section 3.1.5.2, may be implemented in a parameter estimator. There is no change to the structure of the parameter estimation cost functional. The only change to the optimization algorithm is the use of Equation (58) in place of Equation (54). Assuming the ME/I hypothesis and running the optimization algorithm of Section 2.2.1 produces the representative parameter set

$$\begin{aligned}\mathbf{a}^1 &= [17.88]\\ \mathbf{a}^2 &= [35.33]\\ \mathbf{a}^3 &= [53.14]\end{aligned}$$

The cost for a system based on this parameter set is shown in Table 14 along with the corresponding costs for the other applicable systems.

This system was evaluated by running the same type of Monte Carlo simulation as described in Section 3.1.4 using the ME/I hypothesis. The results, tabulated in Table 15, illustrate that optimizing for the ME/I mechanization provides a parameter estimator with lower mean square parameter estimation error on average than other methods using the ME/I mechanization. The simulation results for the optimized ME/I system support the prediction made in the preliminary analysis, as there is good agreement between the predictions in Table 14 and the results in Table 15.

Quantity of Interest	Design Method					
	Meth. 1	Meth. 2	Meth. 3	Meth. 4	Para. Opt.	ME/I Opt.
MS Para. Est. Error ($\mathrm{rad}^2/\mathrm{sec}^2$)	111.52	32.94	169.53	502.75	33.34	32.92

Table 15. ME/I Hypothesis Simulation Results

Note that the ME/I optimized system implemented with the ME/I mechanization exhibits lower mean square parameter estimation error than the best results using the ME/M mechanization (compare Table 15 to Table 13). Recall though, that the ME/I hypothesis is an ad hoc modification to the basic algorithm. The benefit to the example problem, though small, is measurable and could possibly be large in a system where $\boldsymbol{M}$ varies a great deal across the parameter set.

3.2.3 Effect of Probability Lower Bounds on the Parameter Estimator. The simulations performed for the previous section allowed the probabilities to go to zero, to make the simulation match the prediction as closely as possible. Figure 32 illustrates how the parameter converges to an estimate and stays there. The estimate is the parameter on which the selected filter is based. In practice, one would place a lower bound on the probability, as explained in Section 2.2.2, allowing the MMFA to adapt to change in the true parameter. Using the expression in Equation (78) in the parameter estimator cost functional allows us to compute the cost for any given lower bound on the probabilities. Table 16 shows how the cost of the parameter-optimized solution varies with a change in the probability lower bound.

The results of simulations with various values of $p_{\min}$ are shown in Table 17. As was the case with the state estimator example, there is a benefit gained by placing lower bounds on the probability. This benefit, described in Section 3.1.5.4, cannot be calculated beforehand, and outweighs any calculable effects for small values of $p_{\min}$. At larger values of $p_{\min}$, the calculable effects become dominant, similar to the

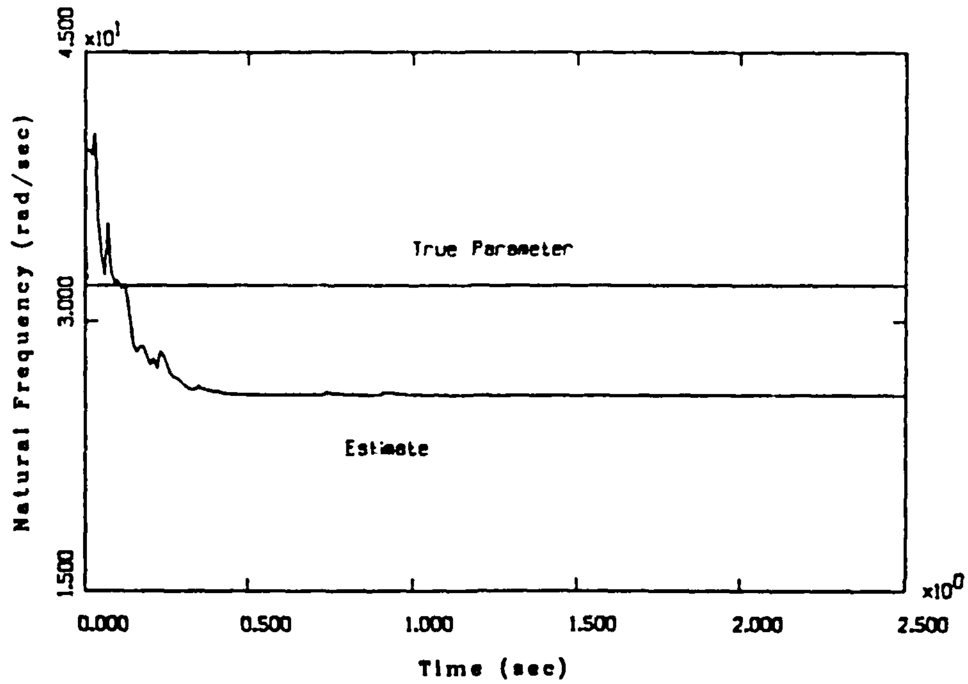


Figure 32. Example Parameter Estimation, $p_{\min} = 0.0$

Probability Lower Bound	J_a^2 (rad^2/sec^2)
0.000	37.04
0.001	37.12
0.005	37.51
0.010	38.07
0.050	45.84
0.100	63.75

Table 16. Cost Variation with Different Lower Bounds on the Probabilities

Quantity of Interest	$p_{\min}$			
	0.000	0.001	0.010	0.100
MS Parameter Estimation Error (rad^2/sec^2)	37.48	35.92	37.57	69.85

Table 17. Bayesian Simulation Results with Various Values of p_{min}

state estimator results (refer to Table 11).

3.2.4 Summary. It is important to note that the optimal state estimator constructed in Section 3.1.2 is not necessarily the best parameter estimator. Indeed, for this example, an MMFA was found which provided parameter estimation with lower average mean square error than any previously accepted method. This was accomplished by minimizing a cost functional which represented the mean square parameter estimation error averaged over the admissible parameter set. Sections 3.2.1 through 3.2.3 illustrate some of the modifications which can be made to the basic MMFA and the corresponding modifications to the optimization algorithm.

Using the ME/I hypothesis and placing lower bounds on the probabilities both provided benefits in the form of lower mean square parameter estimation error. The system optimized for the ME/I hypothesis provides the lowest average mean square parameter estimation error. The benefit from placing lower bounds on the probabilities is not calculable, and outweighs the detrimental effect calculated in the cost functional for lower values of $p_{\min}$ which are likely to be used in an application.

3.3 MMCA Evaluation

In this section, the controller optimization algorithm is used to design an MMCA for the ideal mechanical translational system of Section 3.1.

3.3.1 A Position Regulator. The objective is to design a position regulator for a system with the following state equations:

$$\mathbf{x}^*(t_{i+1}) = \boldsymbol{\Phi}^*(t_{i+1}, t_i)\mathbf{x}^*(t_i) + \boldsymbol{B}_d^*(t_i)\mathbf{u}^*(t_i) + \boldsymbol{G}_d^*(t_i)\mathbf{w}_d^*(t_i) \tag{136}$$

$$\mathbf{y}^*(t_i) = \boldsymbol{C}^*(t_i)\mathbf{x}^*(t_i) \tag{137}$$

$$\mathbf{z}^*(t_i) = \boldsymbol{H}^*(t_i)\mathbf{x}^*(t_i) + \mathbf{v}^*(t_i) \tag{138}$$

The output matrix $\boldsymbol{C}^*$ is set to $[1\ 0]$ to specify position as the desired output. All other matrices are the same as they are in the estimation problem of Section 3.1.

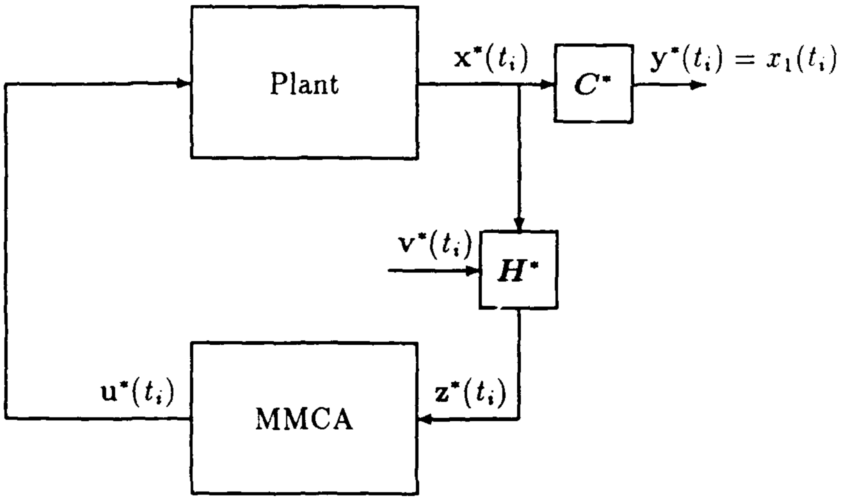


Figure 33. Position Regulator

Figure 33 is a block diagram of an MMCA used as a regulator where the objective is to control the output vector, $\mathbf{y}^*(t_i) = x_1(t_i)$, to zero. The cost functional for this system is written as follows:

$$\mathrm{J}_c^2 = \frac{\int_{\mathcal{A}} \mathrm{E}\{\mathbf{y}^{*\mathrm{T}}\mathbf{W}\mathbf{y}^*\}\mathbf{da}}{\int_{\mathcal{A}} \mathbf{da}} \tag{139}$$

$$= \frac{\int_{\mathcal{A}} \mathrm{E}\{x_1\mathbf{W}x_1\}\mathbf{da}}{\int_{\mathcal{A}} \mathbf{da}} \tag{140}$$

3.3.2 Controller Selection Process. The filters in the bank of controllers are steady-state Kalman filters as used in the estimator problem. The controller gains, $\boldsymbol{G}_c^k$, are calculated using a sampled data Linear Quadratic Gaussian (LQG) stochastic regulator design method as explained in [37, Sect. 14.2]. The gains are calculated to minimize a continuous time quadratic cost functional of the form

$$\mathcal{J} = \int_0^\infty [\mathbf{x}^{\mathrm{T}}(t)\mathbf{S}\mathbf{x}(t) + \mathbf{u}^{\mathrm{T}}(t)\mathbf{U}\mathbf{u}(t)]\mathrm{d}t \tag{141}$$

Minimizing the mean square position autocorrelation corresponds to weighting matrices of

$$\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{142}$$

and

$$\mathbf{U} = [0] \tag{143}$$

These matrices place all of the design emphasis on the position of the system, resulting in high gains on the state feedback without regard to the amount of control energy needed.

Using these weighting matrices and $\mathbf{W} = [1]$, the controller design algorithm described in Section 2.3.2 finds the representative parameter set

$$\begin{aligned} \mathbf{a}^1 &= [25.89] \\ \mathbf{a}^2 &= [41.61] \\ \mathbf{a}^3 &= [56.01] \end{aligned}$$

which is the same parameter set found for the state estimation problem. One might presume that, to have the best controller performance, it is more important to have accurate state estimates than parameter estimates. This is consistent with the idea that a system with full state feedback is more robust than a system which uses state estimates for feedback. Stability (and performance) can be well preserved even with controller gains, G_c^*, evaluated on the basis of incorrect parameter values, as long as the state values are well estimated. Therefore, accuracy in the state estimates is more critical than precision of the parameters needed for gain computation.

3.3.3 Preliminary Analysis. A preliminary analysis was accomplished by calculating the cost functional, J_c^2 of Equation (140), for each choice of parameters. As shown in Table 18, the representative parameter set found by the controller optimization algorithm has the lowest cost. Plots of the steady state position error autocorrelation are shown in Figures 34 - 38 as a function of the truth system parameter for each of the design methods. Triangles on the frequency axis denote elemental controller parameter values. The cost functional was computed by integrating these plots.

Design Method	J_c^2
	(m^2)
Method 1	0.06461
Method 2	0.05530
Method 3	0.07993
Method 4	0.07289
Optimization	0.05076

Table 18. Controller Design Costs

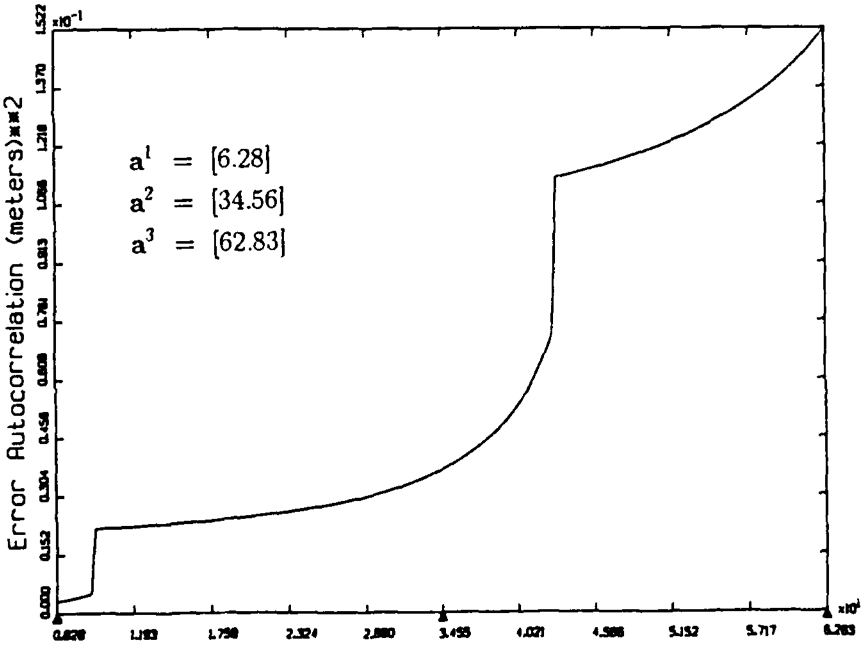


Figure 34. Steady Staie Controller Position Autocorrelation Using Method 1

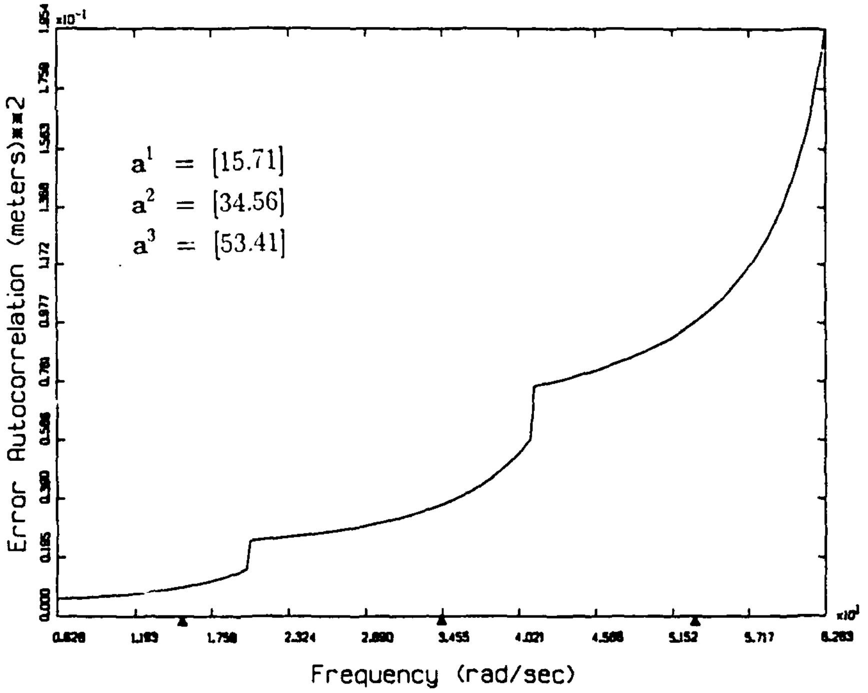


Figure 35. Steady State Controller Position Autocorrelation Using Method 2

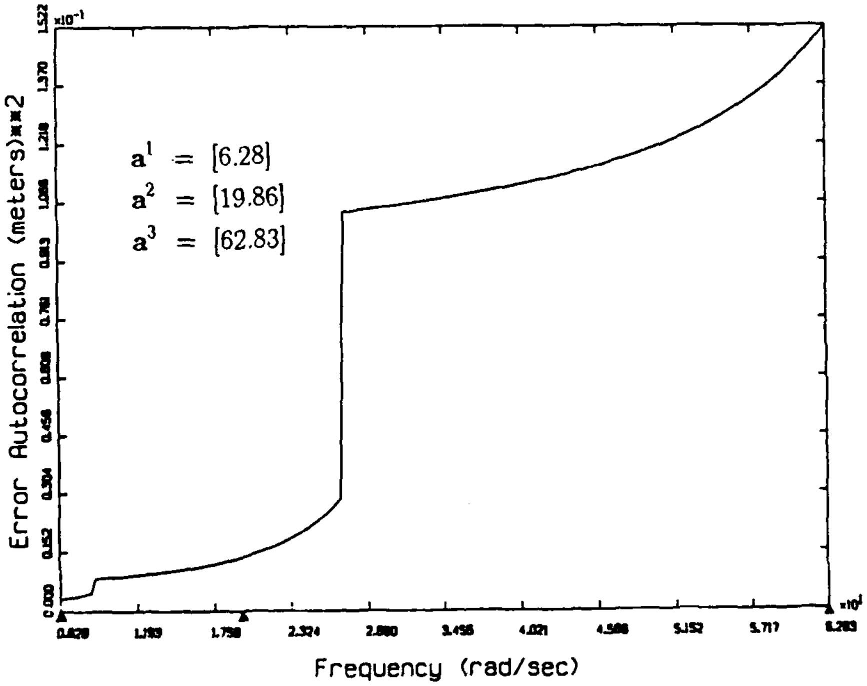


Figure 36. Steady State Controller Position Autocorrelation Using Method 3

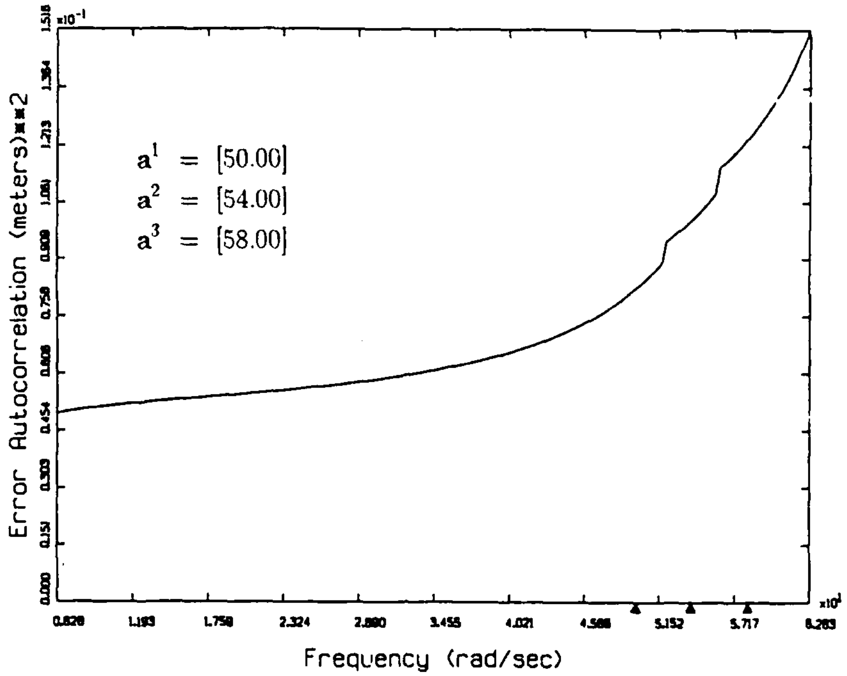


Figure 37. Steady State Controller Position Autocorrelation Using Method 4

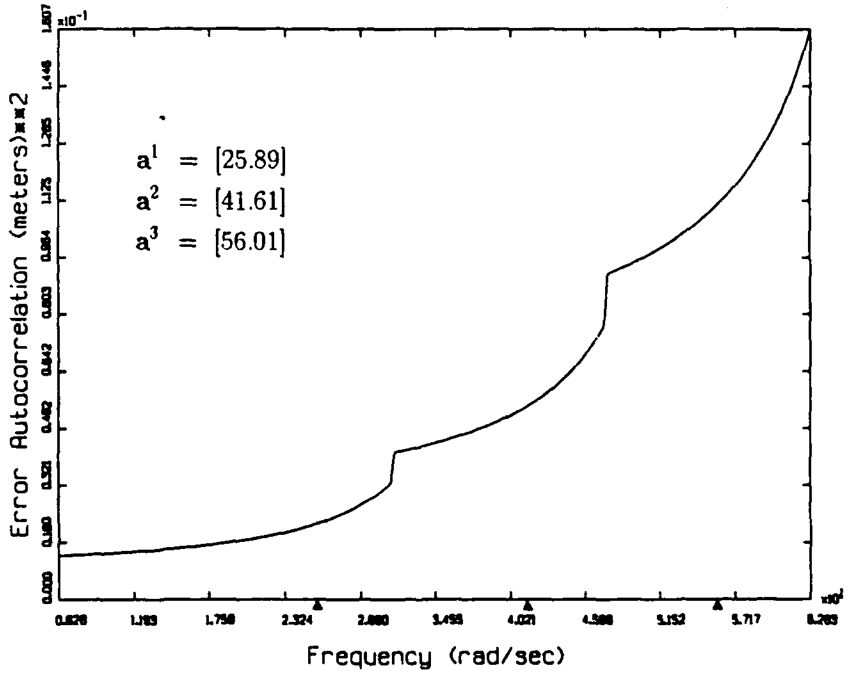


Figure 38. Steady State Controller Position Autocorrelation Using Optimization

The points at which each MMCA switches controllers is determined by the underlying MMFA structure, so the discontinuities in these plots are at the same points as the discontinuities in the plots of the estimator autocorrelations, Figures 16 through 20. For any given controller, the curve is relatively flat at frequencies below the design frequency and increases rapidly at frequencies above the design frequency.

3.3.4 Controller Simulation. To evaluate the parameter discretization found by the optimization algorithm, Monte Carlo simulations were performed, similar to the simulations of Section 3.1.4. A block diagram of the simulation is shown in Figure 39. Though our primary interest was in $\mathbf{x}^*$, the state and parameter estimates were also available for analysis, as were the commanded control values. Equation (127) was used to simulate the truth system with ω_n constant throughout the simulation. Each MMCA design was tested by running 20, ten-second simulations at each of 200 values of ω_n (a^*). The probabilities were propagated by Equation (1) from an a priori probability distribution of $p^1(t_0) = p^2(t_0) = p^3(t_0) = 1/3$, and there was no lower bound on the probabilities. As was the case with previous simulations, the truth model was the same structure as the filter models, and the only difference was the value of ω_n, so the transformation matrix $\boldsymbol{T}$ was the identity matrix.

3.3.5 Results. The results of the Monte Carlo simulations are tabulated in Table 19. The values shown are the average mean square position of the system calculated by

$$\text{Mean Square Position} = \frac{1}{4000} \sum_{\text{runs}=1}^{4000} \{ \frac{1}{800} \sum_{i=201}^{1000} [x_1^*(t_i)]^2 \} \tag{144}$$

The results of the simulation were not the results expected from the preliminary analysis shown in Table 18. The systems resulting from methods 1 and 3 were calculated to be stable across the entire frequency range (see Table 18). As it turns out, they are only marginally stable at certain points in the frequency range, and they

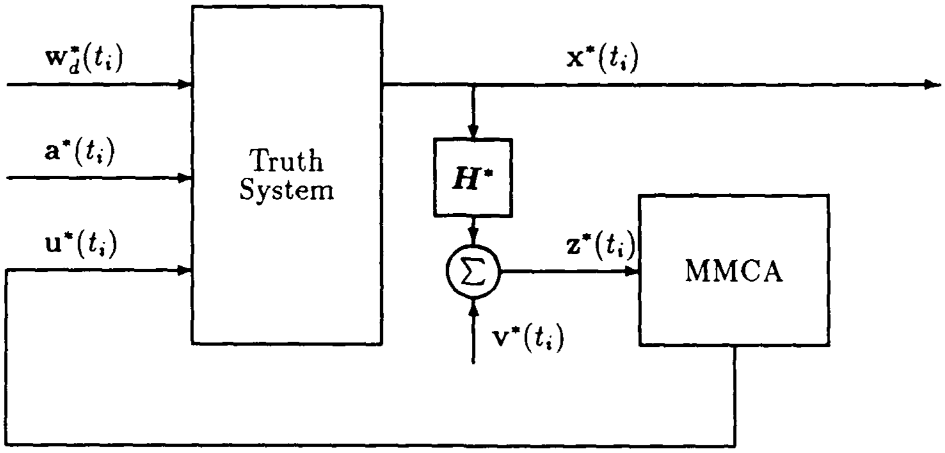


Figure 39. MMCA Simulation Block Diagram

Quantity of Interest	Design Method				
	Method 1	Method 2	Method 3	Method 4	Optimization
MS Position (meters^2)	unstable	0.05543	unstable	0.07312	0.05416

Table 19. Monte Carlo Controller Simulation Results

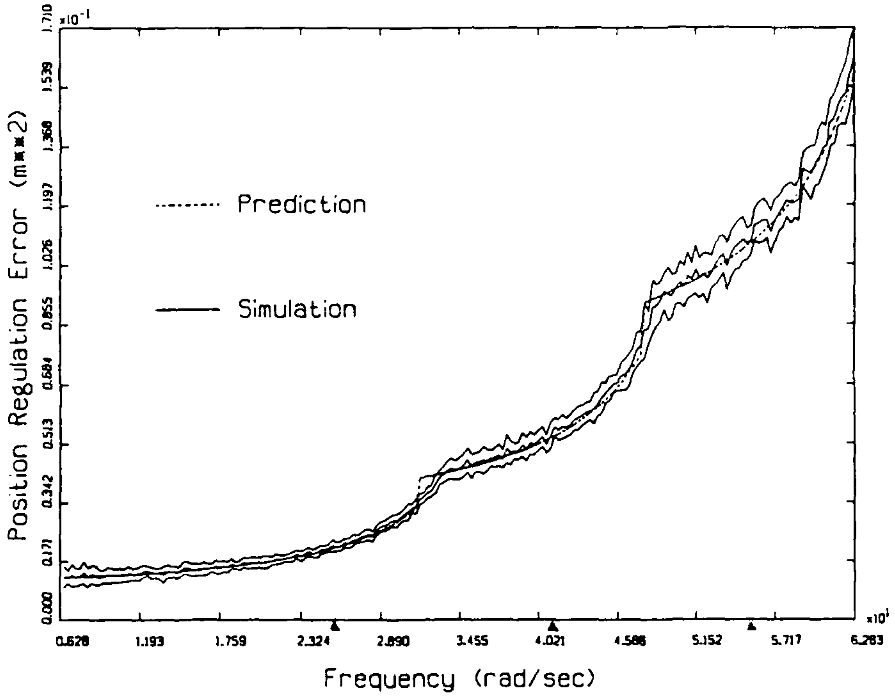


Figure 40. Optimized MMCA Simulation Results, Position Regulation Error

could not maintain stability during the simulations. The results of the simulation of the optimized system are shown in Figure 40. The system performs to within 7% of the mean square regulation error predicted in the preliminary analysis; however, the cost of the method 2 system is within 9% of that of the optimized system cost. Since the difference between the costs of the method 2 and optimized systems is on the order of the accuracy of the simulation, the benefit of optimization is not verified.

By observing the process on a system with more dynamic driving noise, the benefits of optimization become apparent. To illustrate this point, the example problem was changed slightly and the strength, Q, of the dynamic driving noise was increased two orders of magnitude so that:

$$\mathrm{E}\{w(t)w(t+\tau)\} = Q\delta(\tau) = 1.0\ \delta(\tau)$$

This increased energy in the system provides for much more cost differential between systems, and hence, more dramatic results. Since this change affects the filter gains and the residual characteristics, it is necessary to run the optimization algorithm for

Design Method	J_c^2
	(m^2)
Method 1	unstable
Method 2	unstable
Method 3	unstable
Method 4	4.344
Optimization	2.894

Table 20. Controller Design Costs with Q = 1.0

this new problem. The optimization algorithm produces the system

$$\begin{aligned} \mathbf{a}^1 &= [30.64] \\ \mathbf{a}^2 &= [43.85] \\ \mathbf{a}^3 &= [56.39] \end{aligned}$$

The costs computed for this system and the systems found with the other methods are shown in Table 20. The primary impact of the increased dynamic driving noise was to move $\mathbf{a}^1$ and $\mathbf{a}^2$ to the right. With the higher dynamic driving noise, the elemental controllers cannot maintain stability if the true natural frequency is much higher than the frequency assumed within the controller. The algorithm naturally compensates for this. This phenomenon is also the cause for method 2 to be unstable with this higher value of Q when its results are close to the optimized results at the lower value of Q (refer to Table 19).

The systems found using methods 1 through 3 are unstable at some point in the expected frequency range. A designer using one of these methods would assume that more controllers were needed to maintain control over the entire frequency range. This is not the case, as the optimization algorithm found a three controller system that maintains stability over the entire parameter set.

The MMCA is unstable when the eigenvalues of the $\boldsymbol{\Upsilon}$ matrix of Equation (103), are outside the unit circle at some point in the parameter set. In order to keep the

Quantity of Interest	Design Method				
	Method 1	Method 2	Method 3	Method 4	Optimization
MS Position (meters^2)	unstable	unstable	unstable	4.366	2.934

Table 21. Monte Carlo Controller Simulation Results with Q = 1.0

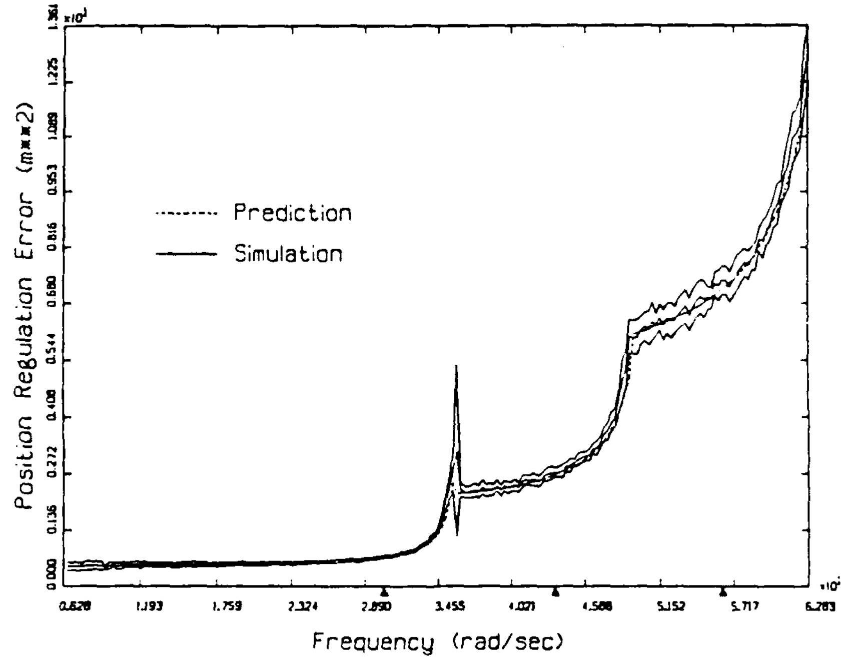


Figure 41. MMCA Simulation Results with Q = 1.0, Position Regulation Error

optimization algorithm viable when it evaluated an unstable MMCA, the cost functional was computed as if the unstable region was not in the admissible parameter set, and a penalty function was added proportional to the size of the unstable region. This allowed the optimization algorithm to move toward an acceptable solution.

All five systems were simulated in the same manner as presented previously, and the results of the simulations are presented in Table 21. The system found by the optimization algorithm performs as predicted, and this is shown in Figure 41. The simulation results are within 1.4% of the predicted mean square regulation error.

3.3.5.1 Effect of The Weighting Matrix on Controller Performance. In Section 3.3.2, a position regulator was designed by placing all emphasis on the position state. If the velocity is also of interest, the output matrix can be changed to

$$C^{*} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

in Equation (137). This opens the door to the use of various weighting matrices. As an example, suppose we keep $Q = 1.0$ and choose

$$\mathbf{W} = \begin{bmatrix} 1 & 0 \\ 0 & 0.01 \end{bmatrix} \tag{145}$$

This matrix places a small additional weight on the velocity.

The controller gains are calculated to minimize a continuous time quadratic cost functional of the form

$$\mathcal{J} = \int_0^{\infty} [\mathbf{x}^{\mathsf{T}}(t)\mathbf{S}\mathbf{x}(t) + \mathbf{u}^{\mathsf{T}}(t)\mathbf{U}\mathbf{u}(t)]\mathrm{d}t \tag{146}$$

To be consistent with the choice of $\mathbf{W}$, we choose

$$\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & 0.01 \end{bmatrix} \tag{147}$$

and

$$\mathbf{U} = [0] \tag{148}$$

Note that this choice produces different gain matrices, G_c^k, than the choice of $\mathbf{S}$ and $\mathbf{U}$ used previously.

Using these weighting matrices, the controller design algorithm described in Section 2.3.2 finds the representative parameter set

$$\begin{aligned} \mathbf{a}^1 &= [25.24] \\ \mathbf{a}^2 &= [41.11] \\ \mathbf{a}^3 &= [62.56] \end{aligned}$$

Quantity of Interest	Design Weight	
	W = diag[1.0,0.0]	**W** = diag[1.0,0.01]
MS Position (meters^2)	2.93	27.49
MS Velocity (meters^2/sec^2)	273793	46303

Table 22. Monte Carlo Controller Simulation Results

Note that these parameters are spread apart farther than the parameters for the position regulator.

This system was simulated in the same manner as the simulations in Section 3.3.4. The results, tabulated in Table 22, show that this system has a lower mean square velocity regulation error at the expense of a higher mean square position regulation error as expected.

3.3.5.2 Using the ME/I Hypothesis in an MMCA. The solution to the basic control problem suggests that the best control results from getting the best state estimates. To improve the state estimation, we used the ME/I hypothesis which chose the filter with the smallest prediction error covariance. It would be a normal assumption that the controller performance could be improved by using the ME/I hypothesis. It would also be wrong for the problem at hand. Figures 42 and 43 show why this is so.

In the estimation problem, with $Q = 1.0$, it would be desirable to shift the switch points to the right so that the filter with the lowest prediction error autocorrelation is chosen (see Figure 42). In the control problem, with $Q = 1.0$, the switch points corresponds to points where the position regulation error autocorrelation is approximately the same for both filters involved (see Figure 43). Shifting the switch points to the right would force the MMCA to select a controller with a higher position

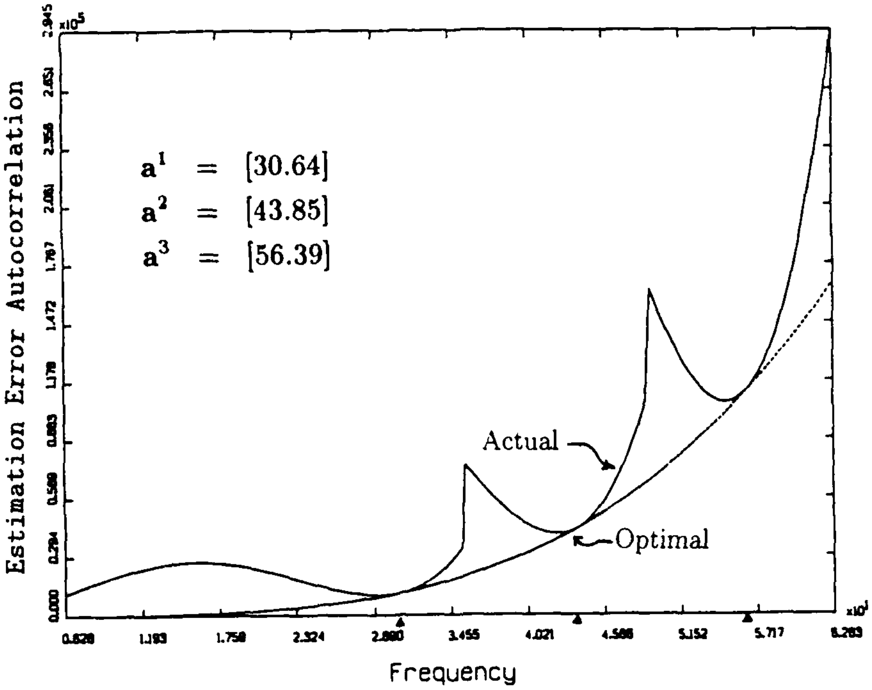


Figure 42. Estimation Error Autocorrelation Using Basic Optimization, $Q = 1.0$

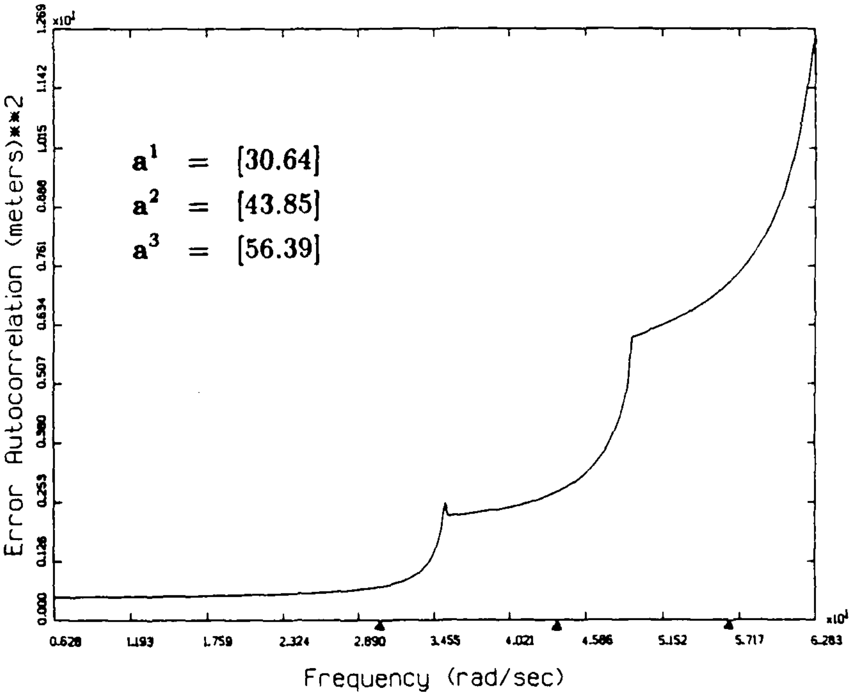


Figure 43. Steady State Controller Position Error Autocorrelation Using Basic Optimization, $Q = 1.0$

regulation error autocorrelation, thus resulting in a larger cost. Calculating the cost of the control system selected in Section 3.3.2 shows that we could expect it to go unstable using the ME/I hypothesis. Even optimizing for the ME/I hypothesis, we cannot find a system with a lower cost than that found with the basic optimization for this problem.

The ME/I hypothesis is an ad hoc addition to the basic MMCA structure. It is shown to be detrimental in the controller example with $Q = 1.0$. This section has been provided to caution the reader in the use of the ME/I hypothesis.

3.3.5.3 Adding a Maximum-Type Cost to the Controller Problem. In a practical design problem, the maximum allowable mean square error may be specified beforehand. In this case, it may be desirable to add a penalty function to the quadratic cost functional, as explained in Section 2.3.3.2, in order to address this specification.

The optimized system for Q = 1.0 has a maximum position regulation error autocorrelation of 12.7 m^2 at ω_n = 62.83 rad/sec. Suppose the designer specifies a maximum mean square position regulation error of 11.0 m^2. Adding a penalty function, like that described in Section 2.3.3.2, and proceeding with the optimization produces the representative parameter set:

$$\begin{aligned}\mathbf{a}^1 &= [31.82]\\ \mathbf{a}^2 &= [45.00]\\ \mathbf{a}^3 &= [57.81]\end{aligned}$$

Compared to the representative parameter set found with the basic optimization algorithm, this set has slightly higher values of $\mathbf{a}^1$, $\mathbf{a}^2$ and $\mathbf{a}^3$, and a little more space between $\mathbf{a}^2$ and $\mathbf{a}^3$. An MMCA designed with this set has a basic cost, J_c^2 of 2.922 m^2 and a maximum regulation error autocorrelation of 10.99 m^2 at ω_n = 62.83 rad/sec (see Figure 44). This system meets the maximum error specification at the expense of average performance.

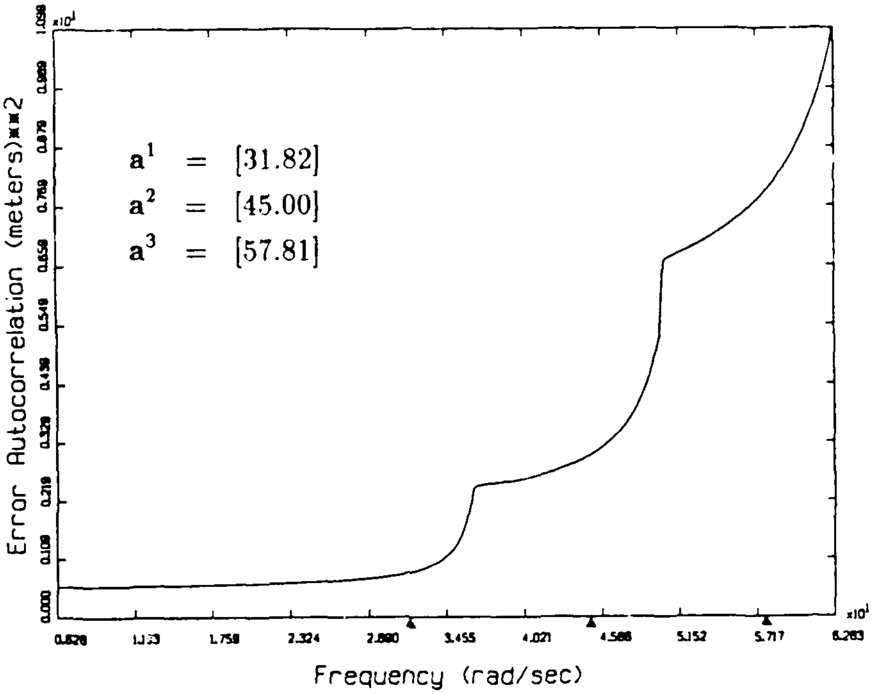


Figure 44. Steady State Controller Position E.ror Autocorrelation Using a Penalty Function

3.3.5.4 Effect of Probability Lower Bounds on Control. In Section 2.3.3.3, the controller optimization algorithm was modified to include lower bounds on the hypothesis conditional probabilities. Using Equation (122) in the controller optimization algorithm enables us to calculate the cost of a system for any given lower bound on the probability. Table 23 shows how the cost of the basic optimized solution varies with a change in the probability lower bound. As $p_{\min}$ is increased, the cost increases, as expected. Note that large values of $p_{\min}$ force the MMCA to use too much of the input from elemental controllers that cannot maintain stability, thus making the entire system unstable.

The results of controller simulations with various values of $p_{\min}$ are shown in Table 24. It is evident from these results that adding a lower bound to the probabilities has the same additional benefit as was shown in the estimator simulations. This benefit is gained by allowing the MMCA to move off a mismatched filter once it has converged, and it cannot be predicted by the cost functional computations.

Probability Lower Bound	J_c^2 m^2
0.000	2.8942
0.001	2.8943
0.010	2.9126
0.050	unstable
0.100	unstable

Table 23. Controller Cost Variation with Different Lower Bounds on the Probabilities

Quantity of Interest	p_{min}			
	0.000	0.001	0.010	0.100
MS Position Regulation Error (meters^2)	2.934	2.931	3.032	unstable

Table 24. Simulation Results with Various Values of p_{min}

However, the detrimental effect predicted by the cost functional computations becomes dominant at larger values of p_{min}, and finally forces the system to be unstable.

3.3.6 Summary. Using a vector minimization routine and an appropriate cost functional, an MMCA was designed which provides the minimum mean square position regulation error when compared with other previously accepted design methods. It was also shown to provide an acceptable control system for a problem in which the three geometric methods do not provide a controller which is stable across the entire parameter interval. This system is shown in simulaıion to provide results very close to values predicted in the design method.

Variations on the original design method are explored in Sections 3.3.5.1 through 3.3.5.4. By manipulating the weighting matrices in the cost functional and control gain design method, it is possible to optimize the control system to provide the minimum mean square error for linear combinations of the state vector components.

It is shown in Section 3.3.5.2 that replacing the ME/M hypothesis with the ME/I hypothesis is not always desirable.

A penalty function can be added to the original controller optimization algorithm, as shown in Section 3.3.5.3, to find systems with a maximum error autocorrelation within specifications. This decrease in maximum mean square regulation error comes at the expense of increased average mean square regulation error.

The ad hoc method of placing a lower bound on the probabilities allows the system to adapt to changing parameters. The calculations required for the optimization algorithm can be augmented to take this into account. The lower bound is predicted to have a detrimental effect on the estimation performance. However, the lower bound is also shown in simulation to have a beneficial effect by allowing the MMFA to move off a mismatched filter after the MMFA has converged to that filter.

IV. Conclusions and Recommendations

4.1 Conclusions

This dissertation is a summary of a research effort into the field of adaptive estimation and control, using a *multiple model filtering algorithm* (MMFA), with a finite number of models, to estimate the states of a system with an unknown parameter vector that can vary over a continuous region of the parameter space. Designing an MMFA involves choosing a finite subset of this uncountably infinite parameter set, and designing filters based on this representative subset. The problem precipitating this research effort is that previously accepted methods for choosing this subset of parameters are ad hoc and provide no means to measure the performance of the resulting system. The goal of this research effort has been the development of an optimization procedure for choosing this discretization of the parameter space.

Toward this end, an algorithm was developed to find a representative parameter set for an MMFA which minimizes a cost functional representing the mean square state prediction error (at t_i^-) averaged over the admissible parameter set. This optimization algorithm was used to design an MMFA for a simple second order problem. Simulations supported the prediction that the representative parameter set produced by the optimization algorithm formed the basis of an MMFA with lower mean square state prediction error on average than any previously accepted method of model selection. As anticipated, this MMFA also exhibited the lowest average mean square state estimation error (at t_i^+). The algorithm is general for any finite dimensional linear time invariant system and includes a means to weight the states separately for cases in which there is a tradeoff between states.

The number of filters, K, in the MMFA is specified by the designer. It was hoped that an efficient procedure could be developed for choosing the number of filters based upon an allowable value of the average mean square prediction error. Unfortunately, the form of the optimization algorithm constrains this procedure to

1. Choose a low value of K for an initial trial.

2. Run the optimization algorithm for this K.

3. If the average prediction error autocorrelation is above the allowable value, increase K and repeat step 2.

4. If the average prediction error autocorrelation is acceptable, stop.

This is hardly an efficient procedure, but it does allow the designer to find the minimum number of filters which provide acceptable performance, assuming that any implementable MMFA can provide acceptable performance.

By replacing the state estimation cost functional with a functional representing the average mean square parameter estimation error, the basic optimization algorithm was modified to find a representative parameter set for an MMFA to estimate the true parameter. The best state estimator is not necessarily the best parameter estimator, as was shown in Section 3.2. The parameter estimation optimization algorithm found a representative parameter set, generally different from the set found in the state estimator optimization, which provided lower mean square parameter estimation error on average than any previously accepted method.

Following the basic development of the estimator algorithm, a separate algorithm was developed to find a representative parameter set for a *multiple model control algorithm* (MMCA), which minimizes a cost functional representing the mean square state regulation error averaged over the admissible parameter set. Through evaluation on a simple second order problem, this algorithm was shown to find a representative parameter set which formed the basis of an MMCA, with lower mean square regulation error on average than any previously accepted design method.

The prediction error autocorrelation is discontinuous at the points in the parameter space where the MMFA can switch from one filter to the next. With the basic form of the MMFA, the estimator switches to a filter with a larger prediction error autocorrelation due to the method of calculating the probabilities within the MMFA. All three algorithms were modified to allow the MMFA to choose the filter with the lowest residual autocorrelation, by including the ME/I hypothesis as described in Section 2.1.3.1. This was shown to be beneficial in the estimator example, and to be detrimental in the controller example with increased dynamics noise strength, $Q = 1.0$. The benefit gained by making this modification is problem-dependent, and should be larger for problems in which the residual autocorrelation, $\boldsymbol{M}$ of Equation (2), changes dramatically across the admissible parameter set.

It was shown that the cost functionals used in this dissertation could be augmented, in two different ways, to include a cost on the maximum mean square error. In the estimator example, an additional cost proportional to the maximum mean square prediction error was augmented to the basic cost functional to produce systems with lower maximum mean square errors. This decrease in maximum mean square prediction error comes at the expense of increasing the average mean square prediction error. In the controller example, a penalty function was added to allow the optimization algorithm to find the best MMCA with a specified maximum mean square regulation error.

Modifications to the basic algorithms were developed to take into account the ad hoc procedure of placing lower bounds on the probabilities. These lower bounds prevent the probability of any filter from reaching zero and removing that filter from further use in the estimation process. For the example problem, these modifications predicted that placing lower bounds on the probabilities would have a detrimental effect on the estimator performance. However, this procedure also has a beneficial effect, which manifested itself during simulation, that could not be predicted by the cost functional calculations alone.

4.2 Contributions

Perhaps the most useful contribution of this dissertation is the procedure for calculating the cost functional. The 'cost functional computation was shown, in

Section 3.1.5, to be a very useful analysis tool in as much as it accurately predicts the performance of the MMFA across the parameter space. Computing the cost functional, and specifically producing plots like Figures 16 through 20, provides a good measure of the quality of any given design. Different MMFA designs, for the same problem, can be compared quantitatively, simply by calculating the appropriate cost functional. Based on this information, one can decide whether or not an optimization is warranted.

The cost functional calculations can also form the basis of an analysis of lower order estimators. The algorithm development of Section 2.1.1.1 includes a transformation matrix, $\boldsymbol{T}$, so that the estimating filter need not be the same dimension as the truth model. By calculating the cost for estimators of various dimension, they can be quantitatively compared.

The controller cost functional calculations can form the basis of a control method comparison. By calculating the cost functional for MMCAs with elemental controller gains calculated by various methods, such as eigenstructure assignment versus LQG, these different design methods can be quantitatively compared.

The main contribution of this dissertation is a formal method for designing MMFAs and MMCAs where none existed previously. This method requires the designer to specify the number of elemental filters or controllers to be used, the weighting matrix in the cost functional, and the method of controller gain calculation. The algorithm can then be automated to find an optimized design.

4.3 Recommendations

The control structure presented in this dissertation is a multiple model regulator in which the state vector is regulated to zero. The modifications necessary for designing trackers, proportional-plus-integral (PI) controllers, or other control laws [37, Chap. 14] were not developed and should be studied in future research. In most cases, this involves augmenting the system models with additional shaping filter states or dynamics states. This does not significantly increase the complexity of the optimization algorithms, but increases the dimension of the problem.

In the preliminary MMCA design, it was found that the best MMCA had the same representative parameter set as the best state estimating MMFA. An analytical study is recommended to see if this is always the case, or if it has something to do with the structure of the example problem.

Previous research [29, 40, 50] brought to light problems with tuning the elemental controllers in an MMCA to increase robustness to unmodelled disturbances. This tuning tends to confound the identification process, making an argument against the application of loop transfer recovery techniques [2] in elemental controller design. An in-depth study of the concept of robustness as applied to multiple model controi systems is necessary so that the effect of robustness recovery techniques on the identification process can be better understood. This could possibly be tied in with an analysis of cost functionals which enhance identifiability, with an end towards designing MMCAs which converge faster, as opposed to MMCAs with optimum average performance.

Bibliography

1. M. Athans and C. B. Chang. Adaptive Estimation and Parameter Identification Using Multiple Model Estimation Algorithm. Technical Report 1976-28, Massachusetts Institute of Technology Lincoln Laboratory, Lexington, Massachusetts, June 1976.

2. Michael Athans. A Tutorial on the LQG/LTR Method. In *Proceedings of the 1986 American Control Conference*, pages 1289–1296, Seattle WA., June 1986.

3. Michael Athans et al. The Stochastic Control of the F-8C Aircraft Using a Multiple Model Adaptive Control (MMAC) Method-Part I: Equilibrium Flight. *IEEE Transactions on Automatic Control*, AC-22 (5):768–780, 1977.

4. Yoram Baram. *Information, Consistent Estimation and Dynamic System Identification.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, November 1976. (AD-A037 972).

5. T. J. Berens Multiple Model Algorithm for In-Flight Simulation. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.

6. C. B. Chang and M. Athans. Hypothesis Testing and State Estimation for Discrete Systems with Finite-Valued Switching Parameters. Technical Report ESL-P-758, Massachusetts Institute of Technology Electronic Systems Laboratory, Cambridge, Massachusetts, June 1977.

7. Chaw-Bing Chang and John A. Tabaczynski. Application of State Estimation to Target Tracking. *IEEE Transactions on Automatic Control*, AC-29 (2):98–109, 1984.

8. J. G. Deshpande, T. N. Upadhyay, and D. G. Lainiotis. Adaptive Control of Linear Stochastic Systems. *Automatica*, 9:107–115, 1973.

9. J. C. Doyle. Guaranteed Margins for LQG Regulators. *IEEE Transactions on Automatic Control*, AC-23 (4):756–757, 1978.

10. Paul G. Filios. Moving Bank Multiple Model Adaptive Algorithms Applied to Flexible Space Structure Control. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1984. (AD-A164 016).

11. R. Fletcher. A New Approach to Variable Metric Algorithms. *Computer Journal*, 13:317–322, Aug 1970.

12. Patrick M. Flynn. Alternative Dynamics Models and Multiple Model Filtering for a Short Range Tracker. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1981. (AD-A115 503).

13. C. S. Greene and A. S. Willsky. An Analysis of the Multiple Model Adaptive Control Algorithm. In *Proceedings of the 19th IEEE Conference on Decision and Control*, pages 1142–1145, Albuquerque NM, 1980.

14. Christopher S. Greene. An Analysis of the Multiple Model Adaptive Control Algorithm. Technical Report ESL-TH-843, Massachusetts Institute of Technology Electronic Systems Laboratory, Cambridge, Massachusetts, August 1978.

15. Louis J. Harambasic Jr. Tracking and Control of a Neutral Particle Beam Using Multiple Model Adaptive Meer Filter. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.

16. R. H. Hashemi, S. Roy, and A. J. Laub. Parallel Structures for Kalman Filtering. In *Proceedings of the 26th IEEE Conference on Decision and Control*, pages 1476–1481, Los Angeles CA, December 1987.

17. R.M. Hawkes and J. B. Moore. Performance Bounds for Adaptive Estimation. *Proceedings of the IEEE*, 64 (8):1143–1150, August 1976.

18. R.M. Hawkes and J. B. Moore. Performance of Bayesian Estimators for Linear Signal Models. *IEEE Transactions on Automatic Control*, AC-21 (4):523–527, 1976.

19. K. P. Hentz. Feasibility Analysis of Moving Bank Multiple Model Adaptive Estimation and Control Algorithms. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1984. (AD-A152 015).

20. Y. C. Ho and R. C. K. Lee. A Bayesian Approach to Problems in Stochastic Estimation and Control. *IEEE Transactions on Automatic Control*, AC-9 (4):*333–338, 1964.*

21. T. Y. Hoballah and P. K. Varshney. Distributed Bayesian Parameter Estimation. In *Proceedings of the 26th IEEE Conference on Decision and Control*, pages 2223–2228, Los Angeles CA, December 1987.

22. Lawrence C. Jamerson. Particle Beam Tracker for an Accelerating Target. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1985. (AD-A163 945).

23. Drew A. Karnick. Moving-Bank Multiple Model Adaptive Estimation Applied to Flexible Space Structure Control. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1986. (AD-A178 870).

24. Drew A. Karnick and Peter S. Maybeck. Moving Bank Multiple Model Adaptive Estimation Applied to Flexible Space Structure Control. In *Proceedings of the 26th IEEE Conference on Decision and Control*, pages 1249–1257, Los Angeles, CA, December 1987.

25. Kenneth K. Kreutz. *Stability and Convergence of Parallel Adaptive Regulators: Towards a Robust Perspective.* PhD thesis, University of California, San Diego, California, 1985.

26. Demetrios G. Lainiotis. Partitioning: A Unifying Framework for Adaptive Systems, I: Estimation. *Proceedings of the IEEE*, 64 (8):1126–1143, August 1976.

27. Demetrios G. Lainiotis. Partitioning: A Unifying Framework for Adaptive Systems, II: Control. *Proceedings of the IEEE*, 64 (8):1182–1198, August 1976.

28. P. R. Lamb and L. C. Westphal. Simplex-Directed Partitioned Adaptive Filters. *International Journal of Control*, 30 (4):617–627, 1979.

29. Robert Lashlee. Moving Bank Multiple Model Adaptive Algorithms Applied to Flexible Space Structure Control. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.

30. Robert W. Lashlee and Peter S. Maybeck. Space Structure Control Using Moving Bank Multiple Model Adaptive Estimation. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 712–717, Austin, TX, December 1988.

31. Thomas A. Leeney. A Multiple Model Adaptive Tracking Algorithm Against Airborne Targets. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.

32. Phyllis A. Loving. Bayesian vs. MAP Multiple Model Adaptive Estimation for Field of View Expansion in Tracking Airborne Targets. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1985. (AD-A155 466).

33. D. T. Magill. Optimal Adaptive Estimation of Sampled Stochastic Processes. *IEEE Transactions on Automatic Control*, AC-10 (4):434–439, 1965.

34. James R. Matthes. Model Selection for the Multiple Model Adaptive Algorithm for In-Flight Simulation. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.

35. Peter S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 1. Academic Press, Inc., New York, 1979.

36. Peter S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 2. Academic Press, Inc., New York, 1982.

37. Peter S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 3. Academic Press, Inc., New York, 1982.

38. Peter S. Maybeck. Adaptive Tracking of Dynamic Airborne Vehicles Based on (FLIR) Image Plane Intensity Data. In N. Christopeit, K. Helms, and M. Kohlmann, editors, *Stochastic Differential Systems: Proceedings of the 3rd Bad Honnef Conference*, pages 284–304. Springer-Verlag, June 1985.

39. Peter S. Maybeck and Karl P. Hentz. Investigation of Moving Bank Multiple Model Algorithms. In *Proceedings of the 24th IEEE Conference on Decision and Control*, pages 1187–1194, Ft. Lauderdale FL., December 1985.

40. Peter S. Maybeck and Donald L. Pagoda. Multiple Model Adaptive Controller for the STOL F-15 With Sensor/Actuator Failures. In *Proceedings of the 28th IEEE Conference on Decision and Control*, Tampa, FL, December 1989.

41. Peter S. Maybeck and Robert I. Suizu. Adaptive Tracker Field-of-View Variation Via Multiple Model Filtering. *IEEE Transactions on Aerospace and Electronic Systems*, AES-21 (4):529–539, 1985.

42. Peter S. Maybeck and William L. Zicker. MMAE-Based Control with Space-Time Point Process Observations. *IEEE Transactions on Aerospace and Electronic Systems*, AES-21 (3):292–300, 1985.

43. D. E. Meer. *Multiple Model Adaptive Estimation for Space-Time Point Process Observations.* PhD thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1982. (AD-A124 827).

44. D. E. Meer and Peter S. Maybeck. Multiple Model Adaptive Estimation for Space-Time Point Process Observations. In *Proceedings of the 23th IEEE Conference on Decision and Control*, Las Vegas, Nevada, December 1984.

45. J.B. Moore and R.M. Hawkes. Decision Methods in Dynamic System Identification. In *Proceedings of the 14th IEEE Conference on Decision and Control*, pages 645–650, Houston Texas, 1975.

46. R. L. Moose, H. F. Vanlandingham, and D. H. McCabe. Modeling and Estimation for Tracking Maneuvering Targets. *IEEE Transactions on Aerospace and Electronic Systems*, AES-15 (3):448–456, 1979.

47. Richard L. Moose. An Adaptive State Estimation Solution to the Maneuvering Target Problem. *IEEE Transactions on Automatic Control*, AC-20 (2):359–362, 1975.

48. Kumpati S. Narendra and Anuradha M Annaswamy. *Stable Adaptive Systems.* Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1989.

49. Allan S. Netzer. Characteristics of Bayesian Multiple Model Adaptive Estimation for Tracking Airborne Targets. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1985. (AD-A163 830).

50. Donald L. Pagoda. Multiple Model Adaptive Controller for the STOL F-15 With Sensor/Actuator Failures. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988.

51. William H. Press et al. *Numerical Recipes.* Cambridge University Press, 1986.

52. R. L. Sengbush and Demetrios G. Lainiotis. Simplified Parameter Quantization Procedure for Adaptive Estimation. *IEEE Transactions on Automatic Control*, AC-14 (4):424–425, 1969.

53. F. L. Sims and D. G. Lainiotis. Recursive Algorithm for the Calculation of the Adaptive Kalman Filter Weighting Coefficients. *IEEE Transactions on Automatic Control*, AC-14 (2):215–218, 1969.

54. Gunter Stein, Gary L. Hartman, and Russ C. Hendrick. Adaptive Control Laws for F-8 Flight Test. *IEEE Transactions on Automatic Control*, AC-22 (5):758–767, 1977.

55. Robert I. Suizu. Enhanced Tracking of Airborne Targets Using Multiple Model Filtering Techniques for Field of View Expansion. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1983. (AD-A141 144).

56. W. Tang and G. L. Mealy. Application of Multiple Model Estimation Techniques to a Recursive Terrain Height Correlation System. *IEEE Transactions on Automatic Control*, AC-28:315–323, 1983.

57. L. D. Tellman and M. B. Leahy Jr. Multiple Model Based Control Development and Initial Evaluation. In *Proceedings of the 28th IEEE Conference on Decision and Control*, Tampa, FL, December 1989.

58. L. D. Tellman and M. B. Leahy Jr. Multiple Model Based Control of Robotic Manipulators: Theory and Simulation. In *Proceedings of the 4th IEEE International Symposium on Intelligent Control*, Albany, NY, September 1989.

59. James S. Thorp. Optimal Tracking of Maneuvering Targets. *IEEE Transactions on Aerospace and Electronic Systems*, AES-9 (4):512–519, 1973.

60. David M. Tobin. A Multiple Model Adaptive Tracking Algorithm for a High Energy Laser Weapon System. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1986. (AD-A178 978).

61. David M. Tobin and Peter S. Maybeck. Substantial Enhancements to a Multiple Model Adaptive Estimator for Target Image Tracking. In *Proceedings of the 26th IEEE Conference on Decision and Control*, pages 2002–2011, Los Angeles, CA, December 1987.

62. G. N. Vanderplaats. ADS - A FORTRAN Program for Automated Design Synthesis. Technical Report NASA-CR-172460, National Aeronautics and Space Administration, October 1984.

63. G. N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design: With Applications.* McGraw-Hill, New York, 1984.

64. Pak T. Yip. Multiple Model Adaptive Filter for Tank Fire Control and its Microprocessor Application. In *Transactions of the Conference of Army Mathematicians*, Bethesda MD, 1982. (AD-P001 033).

65. William L. Zicker. Pointing and Tracking of Particle Beams. Master's thesis, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1983. (AD-A138 311).

Vita

Captain Stuart N. Sheldon [redacted] where he received the degree of Bachelor of Science in Mechanical Engineering, graduating with honors in May 1982. Upon receiving his commission through the R.O.T.C. program he was assigned to the Technical Operations Division (AFTAC) at McClellan AFB in Sacramento California where he served as the Chief, Airborne Engineering Section. He entered the School of Engineering, Air Force Institute of Technology in June 1985. Upon receiving a Master's degree in Guidance and Control as distinguished graduate in December 1986, he immediately entered the Ph.D. program.

[redacted]

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704 0188

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/DS/ENG/89-2

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION
School of Engineering

6b OFFICE SYMBOL (If applicable)
AFIT/ENG

7a NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-6583

7b ADDRESS (City, State, and ZIP Code)

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b OFFICE SYMBOL (If applicable)

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS (City, State, and ZIP Code)

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
An Optimal Parameter Discretization Strategy for Multiple Model Adaptive Estimation and Control

12 PERSONAL AUTHOR(S)
Stuart N. Sheldon, M.S., Capt, USAF

13a TYPE OF REPORT
PhD Dissertation

13b TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT (Year, Month, Day)
1989 December

15 PAGE COUNT
137

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP
01	04	
12	01	

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Adaptive Control Systems, Adaptive Estimation, Multiple Model Adaptive Estimation, Numerical Optimization, Multiple Model Adaptive Control

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Dissertation Cairman: Peter S. Maybeck
Professor of Electrical Engineering

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL
Peter S. Maybeck

22b TELEPHONE (Include Area Code)
(513) 255-3576

22c OFFICE SYMBOL
AFIT/ENG

Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

BLOCK 19

A method is proposed for designing multiple model adaptive estimators to provide combined state and parameter estimation in the presence of an uncertain parameter vector. It is assumed that the parameter varies over a continuous region and a finite number of constant-gain filters is available for the estimation. The estimator elemental filters are chosen by minimizing a cost functional representing the average state prediction error autocorrelation, with the average taken as the true parameter ranges over the admissible parameter set. A second-order example is used to illustrate the increase in performance over previously accepted filter selection methods. By minimizing a cost functional representing the average parameter prediction error autocorrelation, a parameter estimator is designed which is different from the state estimator. The parameter estimator found with this method provides lower average mean square parameter estimation error than previously accepted design methods.

An analogous method is proposed for designing multiple model adaptive controllers to provide stabilizing control in the presence of an uncertain parameter vector. A finite number of constant-gain controllers is used to regulate a system with a parameter vector that varies over a continuous region of the parameter space. The controller elemental filters are chosen by minimizing a cost functional representing the average regulation error autocorrelation, with the average taken as the true parameter ranges over the admissible parameter set. A second-order example is used to demonstrate the improvement in performance over previously accepted controller selection methods. Modifications to the estimator and controller optimization algorithms are investigated which provide the additional capabilities of: changing the filter selection mechanization, using different types of cost functionals, and placing lower bounds on the model probabilities to keep the system open for adaptation.